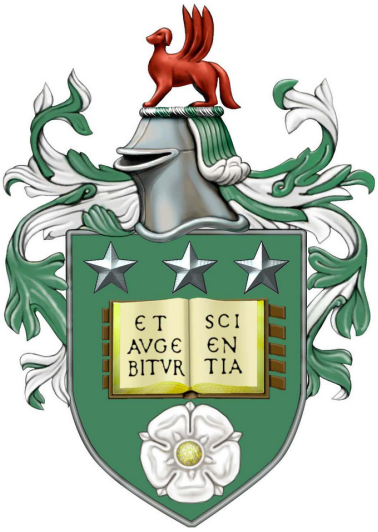
Unsupervised Human Activity Analysis for Intelligent Mobile Robots

Paul Duckworth

Submitted in accordance with the requirements
for the degree of Doctor of Philosophy



The University of Leeds
School of Computing
August 2017

Declaration

The candidate confirms that the work submitted is his/her own, except where work which has formed part of a jointly authored publication has been included. The contribution of the candidate and the other authors to this work has been explicitly indicated below. The candidate confirms that appropriate credit has been given within the thesis where reference has been made to the work of others.

Some parts of the work presented in this thesis have been published in the following articles. The publications are primarily the work of the candidate.

Alomari, M., Duckworth, P.[1]**, Bore, N., Hawasly, M., Hogg, D. C. and Cohn, A. G.** Grounding of Human Environments and Activities for Autonomous Robots. In *26th International Joint Conference on Artificial Intelligence, (**IJCAI**)*, 2017.

Duckworth, P., Alomari, M., Charles, J., Hogg, D. C. and Cohn, A. G. Latent Dirichlet Allocation for Unsupervised Activity Analysis on an Autonomous Mobile Robot. In *Association for the Advancement of Artificial Intelligence (**AAAI**)*, 2017.

Duckworth, P., Alomari, M., Gatsoulis, Y., Hogg, D. C. and Cohn, A. G. Unsupervised Activity Recognition using Latent Semantic Analysis on a Mobile Robot. In *Proceedings of 22nd European Conference on Artificial Intelligence (**ECAI**)*, 2016.

Gatsoulis, Y., Burbridge, C., Dondrup, C., Duckworth, P., and Lightbody, P. QSRLib: a software library for online acquisition of qualitative spatial relations from video. In *Proceedings of 29th Qualitative Reasoning Workshop, at IJCAI*, 2016.

Duckworth, P., Alomari, M., Gatsoulis, Y., Hogg, D., and Cohn, A. Unsupervised Learning of Human Activities by a Mobile Service Robot. *Workshop on Autonomous Mobile Service Robots, at IJCAI*, 2016.

Duckworth, P., Alomari, M., Gatsoulis, Y., Hogg, D., and Cohn, A. A Qualitative Approach for Online Activity Recognition. In *19th Intl. Conf. on Climbing and Walking Robots and Support Technologies for Mobile Machines (CLAWAR)*, 2016.

[1] Joint first author

Hawes, N., Burbridge, C., Jovan, F., Kunze, L., Lacerda, B., Mudrová, L., Young, J., Wyatt, J. L., Hebesberger, D., Körtner, T., Bore, N., Ambrus, R., Folkesson, J., Jensfelt, P., Beyer, L., Hermans, A., Leibe, B., Aldoma, A., Faulhammer, T., Vincze, M. Z. M., Al-Omari, M., Chinellato, E., Duckworth, P., Gatsoulis, Y., Hogg, D. C., Cohn, A. G., Dondrup, C., Fentanes, J. P., Krajník, T., Santos, J. M., Duckett, T., and Hanheide, M.
The STRANDS project: Long-term autonomy in everyday environments.
To appear In *IEEE Robotics and Automation Magazine*, 2017

Duckworth, P., Gatsoulis Y., Jovan, F., Hawes, N., Hogg, D.C. and Cohn, A. G.
Unsupervised Learning of Qualitative Motion Behaviours by a Mobile Robot.
In *Proceedings of International Conference on Autonomous Agents & Multiagent Systems (**AAMAS**)*, 2016.

Acknowledgements

I gratefully acknowledge the financial support provided by the EU STRANDS project (Spatio-Temporal Representations and Activities for Cognitive Control in Long-Term Scenarios) and funded by the European Community's Seventh Framework Programme, Cognitive Systems and Robotics, project reference 600623.

I would like to thank colleagues in the School of Computing at the University of Leeds and in the STRANDS project consortium (http://strands-project.eu) for their valuable input and direction. Finally, I would especially like to thank my supervisors Professors Anthony Cohn and David Hogg for their support.

Abstract

The success of intelligent mobile robots in daily living environments depends on their ability to understand human movements and behaviours. One goal of recent research is to understand human activities performed in real human environments from long term observation. We consider a human activity to be a temporally dynamic configuration of a person interacting with key objects within the environment that provide some functionality. This can be a motion trajectory made of a sequence of 2-dimensional points representing a person's position, as well as more detailed sequences of high-dimensional body poses, a collection of 3-dimensional points representing body joints positions, as estimated from the point of view of the robot. The limited field of view of the robot, restricted by the limitations of its sensory modalities, poses the challenge of understanding human activities from obscured, incomplete and noisy observations. As an embedded system it also has perceptual limitations which restrict the resolution of the human activity representations it can hope to achieve.

In this thesis an approach for unsupervised learning of activities implemented on an autonomous mobile robot is presented. This research makes the following novel contributions:
1) A qualitative spatial-temporal vector space encoding of human activities as observed by an autonomous mobile robot.
2) Methods for learning a low dimensional representation of common and repeated patterns from multiple encoded visual observations.

In order to handle the perceptual challenges, multiple abstractions are applied to the robot's perception data. The human observations are first encoded using a leg-detector, an upper-body image classifier, and a convolutional neural network for pose estimation, while objects within the environment are automatically segmented from a 3-dimensional point cloud representation. Central to the success of the presented framework is mapping these encodings into an abstract qualitative space in order to generalise patterns invariant to exact quantitative positions within the real world. This is performed using a number of qualitative spatial-temporal representations which capture different aspects of the relations between the human subject and the objects in the environment. The framework auto-generates a vocabulary of discrete spatial-temporal descriptors extracted from the video sequences and each observation is represented as a vector over this vocabulary. Analogously to information retrieval on text corpora we use generative probabilistic techniques to recover latent, semantically meaningful, concepts in the encoded observations in an unsupervised manner. The relatively small number of concepts discovered are defined as multinomial distributions over the vocabulary and considered as human activity classes, granting the robot a high-level understanding of visually observed complex scenes.

We validate the framework using, 1) A dataset collected from a physical robot autonomously patrolling and performing tasks in an office environment during a six week deployment, and 2) a high-dimensional "full body pose" dataset captured over multiple days by a mobile robot observing a kitchen area of an office environment from multiple view points. We show that the emergent categories from our framework align well with how humans interpret behaviours and

simple activities.

Our presented framework models each extended observation as a probabilistic mixture over the learned activities, meaning it can learn human activity models even when embedded in continuous video sequences without the need for manual temporal segmentation, which can be time consuming and costly. Finally, we present methods for learning such human activity models in an incremental and continuous setting using variational inference methods to update the activity distribution online. This allows the mobile robot to efficiently learn and update its models of human activity over time, discarding the raw data, allowing for life-long learning.

Contents

List of Figures

List of Tables

List of Algorithms

Abbreviations

CPM	Computational Pose Machine
CDSR	Context-dependent spatial regions
DAG	Directed Acyclic Graph
JEPD	Jointly Exhaustive and Pairwise Disjoint
JPD	Joint Probability Distribution
LDA	Latent Dirichlet Allocation
LSA	Latent Semantic Allocation
LSI	Latent Semantic Indexing
MBR	Minimum Bounding Rectangle
pLSA	Probabilistic Latent Semantic Allocation
QSR	Qualitative Spatial Representation
QTC	Qualitative Trajectory Calculus
QDC	Qualitative Distance Calculus
RGB	Red-Green-Blue
RGBD	Red-Green-Blue-Depth
ROI	Region of Interest
ROS	Robot Operating System
SLAM	Simultaneous Localization and Mapping
SOMA	Semantic Object Maps
STRANDS	Spatio-Temporal Representation and Activities for Cognitive Control in Long-Term Scenarios
TPCC	Ternary Point Configuration Calculus

Symbols

α	Dirichlet hyperparameter
β	Dirichlet hyperparameter
C	(NxM) Term-Frequency Matrix
C_r	Rank r Matrix, low-rank approximation to the term-frequency matrix C
d	Document representation: a sequence of observed graph paths
D	Corpus of observations
Δ	QDC boundary thresholds
η	Maximum graph path length
g	Interval Graph
$\mathbf{h}$	Activity histogram of counts over a code book
i	List iterator (int)
I	Interval representation
I^-	Starting interval set
I^+	Ending interval set
ι	Single interval in an interval representation, defined by lesser and greater endpoints: ι^-, ι^+
ι'	Single node representing an interval in an interval graph
j	Human joint pose (tuple)
k	Integer representing number of centroids in k-means algorithm (int)
M	Matrix column dimension (int)
μ	N-dimensional mean from k-means
N	Matrix row dimension (int)
p	Human body pose (tuple)
ϕ	Topic distribution vector per topic
Φ	Topic distribution matrix
Q	QSR matrix
ρ	Maximum number of entity sets on a graph path (int)
S	Human body pose sequence
Σ	Non-increasing, diagonal, Eigenvalue matrix
t	Human trajectory pose (tuple)
T	Human trajectory sequence
θ	Topic mixing proportion vector
τ	Duration of time (int)
U	Left-singular matrix
V^T	Right-singular matrix
$\mathcal{V}_D$	Code book of code words extracted from corpus D
w^i	Observed graph path
w_i	Unique graph path, a.k.a. Code word

Chapter 1

Introduction

1.1 Motivation

Understanding data from visual input is an increasingly important domain of scientific research. Visual sensors are inexpensive and a natural way to capture similar data that is available to humans through vision. Implementing a learning system on this data aims to mimic the function of the human brain in what can be recognised and understood in the environment. Video cameras capture data about a static scene or a dynamic environment and save this information as images. These images are processed and represented in such a way that a system can extract particular properties or patterns and hence learn about the world being observed. Systems that can understand what activities are occurring in a domain are considered useful for aiding or replacing human tasks. There is a long research history in the security domain where a set of pre-labelled "suspicious" behaviours are learned and recognised. For example a security guard could be alerted to a particular threat on a CCTV monitor without having to constantly watch multiple camera feeds. Allowing the learning system to detect and flag-up suspicious behaviours would allow a reduced number of guards to monitor security.

More recently, activity understanding has become an important area of research in the robotics domain. As computing hardware has improved, the efforts to align robotics and artificial intelligence has increased enormously. Allowing embedded robotic systems to use their own sensors in order to make decisions for themselves allows them to aid humans in a greater range of applications. For example, a robotic car will sense the road using on-board sensors and can apply the brakes itself if it detects an object in the road which it predicts is on course for a collision with. Further, robots that can learn about human behaviour are well suited to operate within human populated spaces. However, unlike security systems that usually monitor the same areas and recognise specific behaviours, robots have to operate in a variety of dynamic environments and situations, often for long periods of time, making autonomous and lifelong learning much more challenging.

Advancements in the reliability of autonomous mobile robot platforms means they are well suited to continuously update their own knowledge of the world based upon their many observations and interactions [Marder-Eppstein et al., 2010, Hawes et al., 2016]. Unsupervised learning frameworks over such long durations of time have the potential to allow mobile robots to become more helpful, especially when cohabiting human populated environments. Such robots can be adaptable to their surroundings, the particular time of day, or a specific individual being observed, saving considerable time and effort hard-coding specific information. For example, a robot butler or house-keeper could learn specific behaviours of its house-guests, with minimal supervision, and perform a specific action given it is observing a learned activity. Understanding what activities occur in which regions and when, allows the robot to adjust its own behaviour, or assist in the task it believes is being undertaken. By removing humans from the learning process, i.e. with no time-consuming data annotation, such robots can cheaply learn from greater quantities of available data (abstracted sensor observations), allowing them to adapt to their surroundings and save time/effort hard-coding specific information.

A key factor for the success of intelligent mobile robots, deployed in human populated environments, is their ability to understand human behaviour and motions. This allows for safer and more effective navigation in populated spaces, and for more useful robotic systems. Such robot systems perceive and represent the world through a range of sensor modalities, often maintaining abstract representations that allow them to make inferences and decisions. How to best represent knowledge about the world is still a major challenge in the field of intelligent robotics. This problem is magnified on mobile robot platforms where on-board sensors provide only a partial and noisy view of the world. The aim is to achieve a robotic system that can observe, learn and gain knowledge over a long period of time from dynamic human populated environments.

In this thesis, we investigate the problem of how an intelligent mobile robot can learn and understand human motion behaviours and simple activities in dynamic real-world human populated environments from partial and noisy observations of the inhabitants. As such, the problem to be solved is to learn from partial human motions and generalise to the underlying human activity behaviours. Figure 1.1 shows an example sequence of images recorded from a mobile robot's field of view, where a person is observed performing a number of simple activities. Ideally, the robot would learn a representation of each activity being observed, in this case each activity is represented as a histogram over a feature space (bottom), allowing the robot to detect new instances of similar activities in the future.

Research Question: Can we learn activities taking place in human populated environments from video data by passive observation from a mobile robot? That is, can we model the qualitative relational interactions taking place between interesting regions of visual data and obtain an activity learning framework similar to functionalities of the human brain? Is it possible to learn models of human activity from a dynamic environment in an unsupervised framework, i.e. without providing the system with feedback on its abstractions?

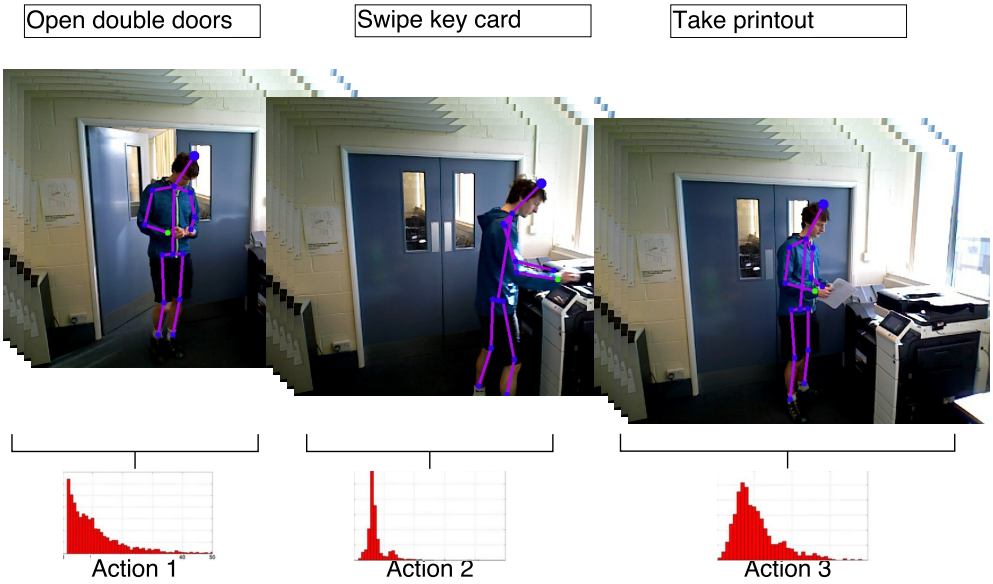


Figure 1.1: An example sequence of images recorded from a mobile robot. Ground truth activity annotation are shown (top), sequence of images overlaid with a human pose estimate (middle), and hypothetical histogram representation of each activity (bottom).

1.2 What is an Activity?

An "activity" can relate to a broad range of different domains such as economic activity, criminal activity or sporting activity. In each of these situations, the term "activity" often relates to a temporally dynamic configuration of some "agents", where the agents can be grounded in the real-world, or could be online agents. There is a considerable amount of work in the literature focusing on activity understanding and the terminology used is not always consistent. Terms such as "activity", "event", "movement", "temporal action/behaviour", "action", "gesture" and "primitive event" are all used to describe similar concepts that can be captured using video data [Bobick, 1997, Cohn et al., 2003, Zelnik-Manor and Irani, 2006, Brémond, 2007]. This is due to the ambiguous definitions of these words in natural language. A mutual set of conditions a so-called "event" or "activity" represented in video data must satisfy [Lavee et al., 2009] is:

1. they must occupy a period of time;
2. they are built out of smaller semantic unit building blocks;
3. they are described using the salient aspects of the video sequence input.

Using this definition, an occurrence of interest in a video sequence is defined as an activity. We therefore distinguish between a general activity, described as an entity which satisfies these three conditions, and a particular activity which has these qualities explicitly instantiated.

Much of the previous learning of activity models has been performed in a supervised learning setting and then the knowledge gained has been applied to an unseen test subject. This is defined as *activity detection*: the act of comparing data with a known or learned activity model and deciding the presence or absence of an activity [Xie et al., 2008]. The proposed research direction into the unsupervised learning setting is defined as an *activity discovery* task. That is, finding activities without knowing their semantics beforehand, using the regularity or self-similarity among multiple activity instances. Automatically detecting previously unknown activities can be very useful when there is a need to explore new environments, or where a lack of supervision is provided and a new observation is unaccounted for among the set of known activities.

An assumption made in this work is that human movements relative to *key objects* that provide functionality in the environment are highly informative of recurrent patterns and relate to everyday activities. For example, standing up and walking from a desk towards the printer to collect a printout, is a human behaviour which occurs in many offices, irrespective of the exact (x, y) locations of both the *desk* and the *printer*. For all offices, it is difficult to generically express this behaviour in the Cartesian map-plane using a quantitative approach. However, it is simple to describe this behaviour in relative terms, across many offices. Qualitative spatial calculi are well suited to this task as they are able to abstract specific details of observations, extracting similarities whilst preserving qualitative differences [Chen et al., 2015]. We therefore focus specifically on human activities that involve objects; we represent an observed human as a sequence of qualitative spatial-temporal relations between the detected human and reference objects within the environment. This allows us to abstract away from absolute Cartesian coordinates into a more structured qualitative space. Qualitative representations are highly tolerant to noisy, partial or varying lengths of observations and therefore suitable for a mobile robot with embedded sensors.

1.3 Main Challenges

A major challenge in this work is that our mobile robot's on-board sensors only grant the system a limited field of view, restricted by its sensory modalities. This translates into a challenging partial and mobile view of the environment, i.e. it obtains obscured, incomplete and noisy human observations. This problem is compounded when the mobile robot is autonomously selecting its own, potentially improper, view points in order to observe its environment. As an embedded system its perceptual limitations restrict the resolution of the human activity representations it can hope to achieve.

A second challenge in the area of human activity analysis, performed in the real-world, is that each observed activity instance is often carried out with particular variations, e.g. opening a door with opposite hands. This is called intra-class variation and in our setting, the robot's observations have large variation between them, even for similar activities. We present no

supervision for each recorded observation and so a major challenge is to learn latent activity models, which are coherent and align well to how humans perceive their behaviours or activities.

Our framework addresses these perceptual challenges in two phases; first by utilising a state-of-the-art human body pose estimator in order to improve the quality of its observations, and secondly we use a *Qualitative Spatial Representation* (QSR), which abstracts quantitative data to a discrete set of qualitative values, thus converting somewhat noisy observations of arbitrary spatial positions into semantic low level actions. This allows the system to compare observations based upon key qualitative features and learn common patterns in an abstracted space, instead of their exact metric details which can arbitrarily differ. This is shown to be somewhat viewpoint invariant, learning a representation of human activities with respect to key locations in the robot's environment. For example, if a person reaches for a mug, the exact spatial position of the hand or mug are not as useful for learning the human activity "making coffee", as a qualitative representation of the hand approaching the mug.

1.4 Novel Contributions

The aim of our work is to understand human activities taking place from long term observation of real-world scenarios. We present a novel unsupervised, qualitative framework for learning human activities in a real-world environment, which is deployed on an autonomous mobile robot platform. The challenge is to learn semantically meaningful human activities by observing multiple people performing everyday activities, and learn a vocabulary which can describe them. The contributions of this thesis are:

- A qualitative spatial-temporal vector space encoding of human activities as observed by an autonomous mobile robot.
- Methods for learning a low dimensional representation of common and repeated patterns from multiple encoded visual observations.
- Two state of the art, long-term human activity datasets recorded from an autonomous mobile robot for the community's use. 1) A six week human trajectory dataset collected along with meta-data such as maps and ROS messages is available at: http://doi.org/10.5518/34. 2) A one week human body pose dataset collected at the University of Leeds, along with meta-data and software repository is available at: http://doi.org/10.5518/86.
- Open source software library contributions, QSRLib for encoding qualitative representations, specifically of time series data [Gatsoulis et al., 2016b,a].

1.5 Overview of Thesis

Our mobile robot is able to autonomously patrol and observe an environment for long durations of time. During this time, it can observe and record many human motions and simple

activities occurring in its environment. Figure 1.2 depicts the robot's observation process from quantitative detections to sparse human activity vector space representation. The flow chart comprises of four sections, which are described in detail in each of their respective chapters. We briefly introduce each of them next.

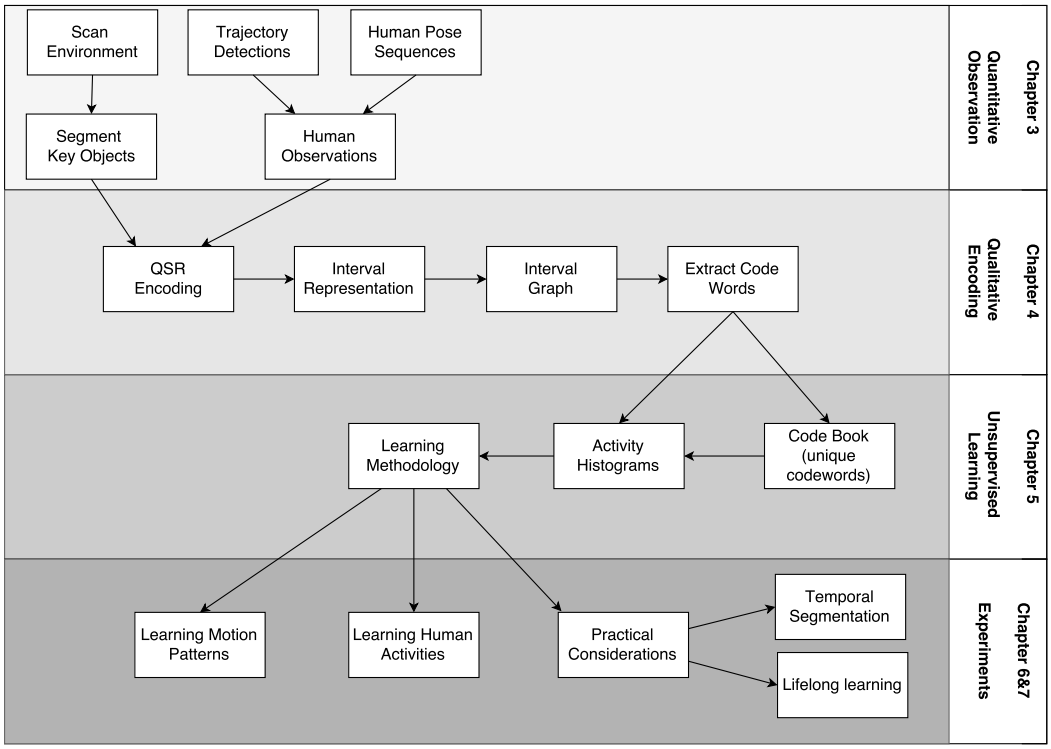


Figure 1.2: Organisational flowchart of the framework presented in this thesis, and how the sections interact.

– Chapter 2: We provide a review of similar literature in the area of activity analysis, with a focus on human activities learned from visual observations and possible visual abstraction methods. Also, we review recent work on intelligent mobile robots performing such learning in dynamic real-world environments.

– Chapter 3: Quantitative observations. The robot detects humans using an RGBD/laser sensors and estimates and tracks their 2D location (human trajectory) or their main skeleton positions (human body pose). It also extracts and encodes key object positions within its environment by performing 3D sweeps of its RGBD sensor.

– Chapter 4: Transformation of the observed pose estimates into a qualitative space. Each quantitative observation is encoded using multiple qualitative calculi in order to abstract the exact Cartesian coordinates. Qualitative intervals are temporally abstracted using

a temporal calculus to compute an *interval graph* where graph paths are extracted and used as *qualitative descriptors* in a vector representation of an observation.

- Chapter 5: Given multiple observations, the unique descriptors form a vocabulary of *code words* and sparse activity histograms are generated for each observation using the counts of each code word extracted. These histograms form a term-frequency matrix which represents the entire corpus of observations. We then introduce multiple unsupervised learning methods, such as simple unsupervised clustering, matrix low-rank approximations and a generative probabilistic approach in order to recover semantically meaningful latent concepts that exist encoded in the obtained feature space. We draw upon simiarities to information retrieval systems which analyse the main topics or themes encoded in a corpus of natural language documents.
- Chapter 6 and 7: Experiments. We present experiments and results from a six week robot deployment at the UK offices of the G4S security company in order to learn human motion patterns. Secondly, more detailed human body pose activities are learned from two datasets, one a publicly available, static camera set-up, and a second, recorded from one week observation from a mobile robot. Finally, we investigate multiple practical consideration which facilitate lifelong human activity learning from a mobile robot. These include, the specific level of manual temporal segmentation applied to each video sequence and incremental learning approaches in order for the robot to perform lifelong learning "in-the-wild".
- Chapter 8: We draw final conclusions and summarise the contributions of the work presented. We also provide multiple possible future research directions.

In summary, the flowchart depicts a learning framework that can: *i*) observe multiple quantitative observations, *ii*) abstract them using qualitative calculi, *iii*) encode the observations as feature vectors, and finally *iv*) perform unsupervised learning relating the concepts extracted to simple human behaviours and activities, with an aim of moving towards lifelong learning "in-the-wild".

Chapter 2

Related Work

Learning models of observed human activity in an unsupervised setting from an autonomous mobile robot is at the intersection of many areas of artificial intelligence research, such as, computer vision, robotics, qualitative spatial representations and information retrieval. In this chapter we review relevant literature relating to these areas and focus specifically on the following aspects:

- What do we mean by human activity analysis and how has it been performed in recent literature?
- How are qualitative abstraction methods used to generalise multiple observations in order to learn common patterns of behaviour?
- What activity analysis is currently performed on intelligent mobile robots and what level of performance has been achieved on similar robotic platforms. How was the problem simplified for the robot platform?
- Learning "in-the-wild", we present works in the literature of unsupervised learning techniques and incremental learning methods, that are considered appropriate for activity analysis on robotic platforms in dynamic, human-populated environments.

2.1 Human Activity Analysis

In the literature, there is a common distinction between vision-based human activity analysis, which extracts information from video and depth cameras using computer vision techniques, and sensor or wearable computing-based systems [Chen et al., 2012, Lara and Labrador, 2013]. Sensor-based systems often rely on the availability of small sensors that can be attached to a human under observation in order to obtain a representation of a person's movements or behaviour, namely wearable sensors, smart phones, or radio frequency identification (RFID)

tagged objects. They have recently obtained good performance when coupled with Deep Learning techniques [Ravi et al., 2016, San et al., 2017] or Hidden Markov Models [Trabelsi et al., 2013], however, from the perspective of an autonomous mobile robot in a real-world environment, it is not practical to equip every observed person with such sensors in order to learn about their behaviours. Instead, we focus on vision-based activity analysis, which is more suitable for a passive robotic system. The following section describes previous work in the literature relating to the task of human activity analysis from vision.

Much of the work in computer vision literature is not specifically interested in learning activity models from video data. Instead hand-defined models are used and the task is reduced to activity discovery, such as in [Ivanov and Bobick, 2000, Medioni et al., 2001, Ryoo and Aggarwal, 2009, Morariu and Davis, 2011]. Hand-defining activity models does not scale or generalise well to an autonomous mobile robot framework operating in dynamic environments. This section is organised using the concept that understanding activity in video data is comprised of the following two sub-tasks [Lavee et al., 2009]:

- Abstraction: The task of translating video sequence inputs into intermediate units amenable by activity models. Popular methods include pixel-centric or object-centric approaches.
- Activity modelling: A sub-domain of activity understanding, devoted to learning a formal description of activities of interest. The inputs from the abstraction layer are used to determine whether such an activity has occurred in a particular video sequence.

We describe both in more detail next.

2.1.1 Visual Abstraction Methods

An important challenge is to obtain video data that accurately represents the environment. Three common methods of capturing 3D video data from sensors are: 1) using a state-of-the-art marker-based motion capture system such as MoCap[1]; 2) using stereo cameras to capture 2D image sequences from multiple views in order to reconstruct 3D information about the environment [Argyriou et al., 2010]; 3) using a relatively inexpensive RGBD depth cameras that provide access to registered 3D point-cloud video data at high frame rates and resolutions. The reader is pointed to [Aggarwal and Xia, 2014] for a review of human activity analysis specifically from the various 3D video techniques. In this thesis, we focus on using RGBD depth cameras in order to obtain an accurate visual understanding of the environment, specifically because they are small, cheap and practical to mount on to a mobile robot. Two such structured light depth sensors are Microsoft's Kinect [Kinect camera [online], 2017] and the ASUS Xtion Pro-Live sensor [Xtion ASUS camera [online]]. They both consist of an infrared (IR) projector, an IR camera, and a colour camera. The IR projector casts an IR speckle dot pattern into the 3D scene while the IR camera captures the reflected IR speckles.

[1] http://mocap.cs.cmu.edu/.

Understanding activities taking place in video data relies upon obtaining low-level aspects of the images and translating them into meaningful activity descriptions. These are often described as *features*, where good features have the ability to provide useful abstract properties about the content of a video to the system. There are three types of invariance, so-called good features should have: view-point invariant; execution-rate invariant and anthropometric invariant [Sheikh et al., 2005]. Two techniques to extract features from images commonly used are pixel-based abstraction and object-based abstraction, each are described next.

Pixel-based abstraction This abstraction method relies on pixel-based information extracted from images such as colour, texture or gradients. The focus is to model the semantically meaningful relationships between neighbouring pixels by taking their various values. Histograms of gradients (HOG descriptors) is a popular technique in the literature, especially for detecting humans in images [Dalal and Triggs, 2005, Zhu et al., 2006, Yang et al., 2012]. A second popular pixel-based abstraction method is to compute Spatial Temporal Interest Points [Laptev, 2005]. This representation can be obtained by taking multiple patches of the image across multiple frames (over time) around detected interest points, specified for instance by using the Harris corner detector [Harris and Stephens, 1988].

Different methods have been applied to the extracted patches in order to compute features that represent local structures in space-time where the image values have significant local variations in both space and time; commonly, (*i*) histograms of oriented (spatial) gradients (HOG) and (*ii*) histograms of optical flow (HOF) are used [Dollár et al., 2005, Savarese et al., 2008, Laptev et al., 2008]. Contextual features can also be added to the descriptor to incorporate the spatio-temporal dependencies into a bag-of-words model [Schuldt et al., 2004, Wang et al., 2011].

Using multiple different temporal resolutions, HOG descriptors have been shown to be able to recognise periodic/non-periodic activities, isolated occurrences/repetition, structured video and dynamic textures [Laptev, 2005, Zelnik-Manor and Irani, 2006]. Another pixel-based abstraction technique uses motion energy images (MEI) which are generated from binary cumulative motion images. These can be extended into motion history images (MHI) by implementing a function which models the temporal history of motion [Bobick, 1997, Bobick and Davis, 2001]. The result is a scalar-valued image where more recently moving pixels appear brighter.

Modelling relationships between the very low-level pixel values is often very efficient, however, they can provide an insufficient spatial representation of a human body to capture structurally complex human activities. For example, approaches assume all motion in the scene should be incorporated into the temporal template. This is the main limitation with pixel-based abstraction methods and why they are not well suited to represent multiple types of motions or activities which involve interactions between multiple objects, people, and/or a mobile camera frame. Secondly, pixel-based approaches lack more long-term temporal information

which means that more complex activities, that can not be represented by simple repetitive patterns, are difficult to accurately represent, or for an intelligent system to learn.

Object-based abstraction It is often easier to reason about the physical world by first using a robust segmentation algorithm for detecting physical objects of interest. In object-based abstraction techniques, low-level input data is first abstracted into a set of entities or objects with corresponding properties. Common properties in the literature include position, speed and trajectory. Minimum bounding rectangles (MBRs) and blobs are popular object-based representations [Fernyhough et al., 1998, Hongeng and Nevatia, 2001, Sridhar et al., 2010, Cohn et al., 2012, Dubba et al., 2015]. Reasoning about MBRs in an object-centric method works well for high-level reasoning, where multiple interacting objects can be represented as performing different spatial arrangements. However, object-based abstraction techniques rely on good detections and tracking of the corresponding objects of interest, which is a challenging sub-domain of computer vision in itself. Further, MBRs do not extend well to complex shapes, such as those obtained from observing a person's body pose. Silhouettes are a second object-based technique and are sometimes described as space-time shape features [Gorelick et al., 2005]. However, using silhouettes relies on accurately segmenting the background and assuming that people, objects and activities can all be recognised from the outlines of their shapes, textures or motions which is not always the case.

As well as MBRs, commonly the position of the human body is estimated from images and is used in object-based abstraction methods. The idea is that if a system can obtain accurate estimates of the positions of a person's body joints, it can learn a detailed representation of the motion of specific body joints over time/frames. A popular approach to obtain human pose estimates is to use the Microsoft's Kinect sensor and software [Kinect camera [online], 2017]. This was originally developed as an interface with the Microsoft Xbox games console [Microsoft Xbox, 2017], and allows interactions between users and a game without the need to touch a controller. It allows for up to six people to be recognised in a scene and their body pose inferred, which has been very useful in the field of computer vision; a review of work carried out using this method is presented in [Han et al., 2013]. The sensor was developed for gaming, and the tracking is optimized to recognize users standing or sitting, and facing the sensor; sideways poses provide some challenges regarding the part of the user that is not visible to the sensor. A further limitation is that the software is not compatible with the Robot Operating System [Quigley et al., 2009] used by many robots. An alternative method, and one that we use, is to use the ASUS Xtion Pro-Live [Xtion ASUS camera [online]] with the OpenNI SDK for person tracking [OpenNI organization, 2016] which uses depth-only data to detect people and infer their 3D pose in real-time.

More recently, the human body pose can be very accurately inferred from RGB images using pre-trained Neural Networks, such as in, [Charles et al., 2014, Pfister et al., 2015, Wei et al., 2016]. We use a Convolusional Neural Network in a post-process step on images where a

person is detected (using OpenNI) in order to improve the human body pose estimate, however it is more computationally demanding and requires full access to the robot's GPU. Further, view-point invariant body pose estimates have been achieved by embedding local regions of depth images from multiple sensors into a view-invariant feature space, and a recurrent and convolusional neural network is used to accurately infer the human body pose [Haque et al., 2016].

2.1.2 Activity Modelling

Given a visual abstraction method, a second sub-task is learning a representation of an activity that is taking place in a video. In general, information is extracted from the video and abstracted to be used as input to an activity learning framework. The specifics of the learning framework often depend upon the domain and the exact visual abstraction method chosen. The techniques used in the literature loosely fall into the following categories: pattern-recognition methods, state models, logic or grammar models. They are discussed in more detail here, borrowing ideas from activity analysis survey [Lavee et al., 2009] and literature reviews in [Brémond, 2007, Sridhar, 2010].

Pattern-recognition methods These methods focus on a regular pattern recognition/classification problem and usually relate to identifying simple types of human motion such as *walk, run, jump* and *skip*. These techniques often use low-level visual abstraction methods that are well suited to model motion within a sequence of images which are often recorded by static camera set-ups that contain a single human moving in the images. These classifiers are generally well understood, where the activity learning is often fully specified by providing many training samples, and new video instances can be classified into the learned model using a measure of distance. There are many examples in the literature which use pattern-recognition techniques such as nearest neighbours with motion history images [Bobick and Davis, 2001] or with silhouettes [Gorelick et al., 2005]; Support vector machine (SVM) using a recursive filtering abstraction similar to MHI [Cao et al., 2004]; boosting [Laptev and Pérez, 2007]; and neural networks [Yang et al., 2008, Donahue et al., 2015].

State models Models which combine human intuition about activity structure with machine learning techniques are often described as state models. These models use semantic information to specify the state space of the activity model. The semantic information associated with the model structure makes these models difficult to learn from training data. However, once the model structure is specified, model parameters are learned from the training samples. State modelling formalisms include finite state machine (FSM) also known as finite-state automata, which are fully observable and able to model sequential aspects of video events in a relatively simple way from provided training samples. They have been used to model hand gestures from low-level image features [Jo et al., 1998], single-actor actions using silhouette matching [Lv

and Nevatia, 2007] and multiple person interactions from characteristics of the trajectory and moving blobs [Hongeng and Nevatia, 2001].

State models have been extended to utilise probabilities in order to model the uncertainty of observations and interpretation in video data. For example, semantic knowledge of a domain can be input into a Bayesian Networks (BN) (state space) and can be factorized into variables of specific interest. Formally, BN are graphical models where nodes represent random variables and directed edges represent conditional independence. Given the joint probability, inferences can be made about any of the latent variables in the model, by using the observed data. They have been used with pixel-based abstraction methods to recognise multi-agent activity patterns of human-motion blobs in surveillance images [Hongeng, 2004], and also with spatio-temporal interest points to learn a representation of various action categories [Niebles et al., 2008]. These works share similarities to the probabilistic generative model we describe in Chapter 5, however, they use low-level image features whereas our qualitative abstraction allows more long-term temporal information to be encoded in each observed random variable.

Other popular state models used in the literature include: Hidden Markov Models (HMMs) for sign language and gesture recognition [Schlenzig et al., 1994] and identifying single-person view-invariant actions [Ogale et al., 2004]; dynamic BNs (DBNs) to encode a duration in each state and learn patterns of moving MBRs in images [Hongeng and Nevatia, 2003]. However, these methods also use low-level image features which do not extend to longer temporal human activities. Lastly, Conditional Random Fields (CRFs) remove the required prior distribution or independence assumption on training instances and have been used to learn natural language patterns from sequential data [Lafferty et al., 2001], although they often require more training time.

Logic models These models provide a framework for learning logical rules relating to activities of interest. They are able to explicitly specify complex, high-level semantic properties such as information about sub-activity ordering, and complex temporal, spatial, and logical relations among such sub-activities. As a result, they can model composite activities in terms of structures of atomic activity, and thus the activities can be somewhat structurally complex. Logical and relational language models are more expressive than the above state space methods as they can encode complex propositions, functions and quantification. The extra expressiveness in these models means that they are usually fully supervised using domain knowledge. Examples of these models in the literature include Petri Nets (PNs) for interpreting activities in surveillance images [Borzin et al., 2007], grammar models and Inductive Logic Programming (ILP) for learning domain specific axioms in a blocks world [Moyle and Muggleton, 1997] and more complex multi-object activities using an object-based abstraction method [Dubba et al., 2010, 2015].

2.1.3 Learning Setting

Work relating to activity modelling can be further split by the nature of the supervision assigned to the training samples. This is known as the learning setting and can be either supervised, unsupervised, or semi-supervised. The setting refers to the nature and degree to which expert knowledge is encoded into the activity modelling framework.

Supervised setting This refers to a range of learning supervision, where at one end of the scale is fully manually defined hand-crafted rules to describe the activities of interest. A more standard supervised learning setting however is where each training sample contains a single activity instance and is assigned a ground truth annotation specified by a domain expert, known as its "label". Then, during a training phase a model is fitted using multiple data-label training sample pairs. The aim is to learn an activity model that separates the training samples based upon their labels, then classification of new instances is performed by using a distance measure to the model components. However, obtaining manual supervision for every observation can be a labour-intensive task, that is, first temporally segmenting video sequences into activity instances and then manually annotating each. Further, supervised temporal segmentation of video sequences can often be ambiguous, e.g. the exact frame a specific activity starts or ends can often be undefined and the ground truth may vary between experts. The majority of research conducted in the activity modelling literature is in the supervised learning setting. One approach to avoid full supervision however, is to create multiple *bags* of instances where at least one is of interest, the bag is then manually annotated [Gu et al., 2016]. This is regarded as weekly-supervised learning or multi-instance learning, where one label corresponds to multiple instances for convenience.

Unsupervised setting In principle this setting implies that the obtained sensory data is not enriched by any additional expert information, however within this setting there is again a range of supervision. Commonly, the aim is to separate the unlabelled training samples into representative clusters that contain "similar" instances. It is often assumed that activity instances are temporally segmented from video sequences, but unlabelled. Features are usually extracted and used to represent each instance and clustered using machine learning techniques. The challenge is extracting discriminative features between the different activities in order to obtain separate clusters in the feature space. The unsupervised *discovery* setting is more challenging, where neither the segmentation of the video sequence into activity instances is given, nor is any representation of the activities themselves.

Semi-supervised setting Obtaining manual training annotation can be challenging or time-consuming. One approach is to initialise an activity model using a set of manually annotated training samples (supervision), then update this model or refine it using large amounts of readily available unlabelled training samples in an unsupervised fashion. An example semi-supervised

approach on a vision-tracked indoor activity dataset uses a manual Propagation Network (a form of a DBN) which can be specified and initialised then refined by an unsupervised Expectation Maximisation (EM)-based method [Shi et al., 2006]. Semi-supervised learning is also common for detecting atypical events or anomaly detection using wearable sensors [Zhang et al., 2005, Stikic et al., 2008] where unlabelled data is abundant.

Reinforcement learning (RL) In the RL setting there is often an agent which receives some inputs from its environment, it performs an action and receives a reward based upon its new state within its environment. Crucially, there is no external supervision but there is a defined reward signal (which can often be delayed). One key distinction is that an agent's actions affects the subsequent data it receives, and therefore time matters and observations are not considered sequential, i.e. non-independent and identically distributed. Inverse RL, commonly known as Inverse Optimal Control, is where no reward function is given and the aim is to learn this function from expert observations. It has previously been used to learn motion patterns through *static* scenes and defined as Activity Forecasting in [Kitani et al., 2012].

2.1.4 Learning Human Activities

2D Motion Behaviours

There is a long standing field of research in video surveillance, where it is important to be able to track human movements and behaviours in a particular area being observed by video cameras. Learning patterns of different human behaviours and then recognising new instances of them is an obvious extension to the surveillance domain, and a number of approaches to predict human motion behaviours have been developed. This can be thought of as learning motion patterns in a 2-dimension image plane, and many statistical approaches have been applied in the literature [Johnson and Hogg, 1995, Vasquez and Fraichard, 2004, Hu et al., 2006, Basharat et al., 2008]; along with neural network approaches [Johnson and Hogg, 1995, Hu et al., 2004]; unsupervised clustering techniques [Piciarelli et al., 2005, Luber et al., 2012]; and goal-based state machines [Dee and Hogg, 2004].

The key difference to our framework is that these works use video sequences collected from static cameras with a fixed frame of reference and a wide field of view. This allows them to observe long and complete trajectories across the image plane, often fully observing the motion of each human. This leads to common "entering" and "exiting" locations which can be learned and humans motion can be predicted between these, along common trajectory paths. These static camera approaches make no predictions outside their field of view and therefore have limited use outside a surveillance setting, such as in mobile robotics where the field of view varies as the robot moves, is much more narrow and often occluded. Similarly, motion behaviours over a transportation network have also been learned and predicted from GPS data [Liao et al., 2007]. Here, similar techniques to the surveillance setting can be used since a

person's motion is observed over the entire network. They make no predictions outside of the field of view, and therefore can be considered similar to static camera approaches.

The surveillance domain has recently been extended to include entire homes containing a suite of static cameras/sensors in order to learn behaviours of the occupants, i.e. in a smart home setting. Activities of daily living have been learned from static fish-eye cameras placed into homes of elderly people in order to learn patterns and ultimately create a "summarization for eldercare video monitoring" [Zhou et al., 2008]. Multi-modal activity analysis has been performed using smart home sensors and a suite of wearable sensors for the purpose of an assistant medical robot [Bruno et al., 2015]. Activities have also been learned and discovered in an unsupervised setting from their movements around a smart home [Chen et al., 2016], which uses a similar generative learning framework to our work. However, all these works assume they fully observe the environment and there is only a single occupant in the home. This hugely simplifies the visual/sensor-based abstraction methods and these techniques do not extend to mobile robot camera frames or to obscured camera scenes.

3D Human Body Pose Activities

Activity recognition using the human body pose extracted from visual data is a mature subfield of activity analysis. Usually, the aim is to not only keep track of 2D moving objects in a scene, but also to draw conclusions into what they might be doing. More specifically, human activity recognition aims to understand what action a person is performing in the observed scene. There have been many approaches to this task; the majority use data collected from static RGB cameras, but also more recently from RGBD depth sensors. The reader is pointed to survey papers which cover the topic in detail using RGB cameras [Turaga et al., 2008, Lavee et al., 2009, Weinland et al., 2011] and 3D depth cameras [Ye et al., 2013, Aggarwal and Xia, 2014]. However, many of the common techniques in these surveys perform supervised learning, where each training sample requires manual hand annotation with a ground truth label. This is not a feasible solution for our long term autonomous mobile robot which ideally, has as little supervision as possible. Another key difference to our work, is that a human activity recognition system deployed on a mobile robot has a changing field of view. This presents a challenging and partial view of an environment, making the observations of similar activities vary greatly. One way of addressing challenging view points is to learn correspondence between multiple synthesized views in order to "hallucinate" action descriptors corresponding to potentially unseen viewpoints [Gupta et al., 2014]. However, they first use the MoCap data in order to accurately learn motion exemplars which are then input into a feature mapping for each view change. In our work, we abstract visual observations in order to generalise them and learn patterns in this abstracted feature space, without relying on sensor-based methods.

It has been shown that simple human activities for daily living can be learned using only the abstraction of the human body pose estimates [Sung et al., 2012, Parisi et al., 2015, Cippitelli et al., 2016]. However, these works learn representations of body pose movements, the visual

data is recorded by a static video camera, where activities are clearly staged and repeated multiple times, i.e. it does not adequately represent real-world environments. Shape analysis of the human pose across time has also been used to decompose a full motion sequence into short temporal segments representing elementary motions [Devanne et al., 2017]. The estimated body joint angle positions of a human body pose can be projected into a subspace, and compared across multiple frames in a system that learns exemplars for each action in order to classify new instances [Sheikh et al., 2005]. However it requires the exemplars of each action in a supervised manner unlike our approach.

More interesting and varied human activities can often be represented when the human body pose and MBRs of objects are combined into one representation [Koppula and Saxena, 2013, Hu et al., 2014, Tayyub et al., 2015]. In this thesis, we also use this approach to represent 3D human body pose activities. We represent human behaviours between the estimated human body joint locations relative to the positions of key objects of interest in real-world, dynamic environments.

2.2 Abstract Representations

Spatial and temporal information represented qualitatively is natural and efficient. It provides an abstraction to precise, numerical or quantitative information which can prove unnecessary or intractable. This is the idea behind using a qualitative abstraction in order to represent features extracted from visual sensor data. The key to a qualitative representation is not only that it is symbolic and utilises discrete quantity spaces, but that the distinctions made in these discretisations are relevant to the behaviour being observed and modelled.

2.2.1 Qualitative Representations

It is clear that qualitative information aligns well with how humans reasons about space in the real world and it is believed there are dedicated areas of the brain to perform such abstractions [Amorapanth et al., 2010]. It is therefore natural to attempt to embed this into systems to understand human behaviour in video data and ultimately, into autonomous robotic systems in order to represent dynamic human behaviours in the populated environments they inhabit.

Qualitative spatial and temporal calculi arise from a set of jointly exhaustive and pairwise disjoint (JEPD) relations. There are many types and applications developed in the literature, some of the most popular include topological, directional and non-topological, i.e. distance, motion, size and shape [Chen et al., 2015]. Qualitative representations are often used to represent quantitative observational data in a low-dimensional and more semantically meaningful qualitative space, as in this thesis. They have also been used to check the consistency of observations and remove noisy video sequences [Fernyhough et al., 1998]. However, they are also used extensively for qualitative reasoning tasks and applied to many real world domains [Cohn et al., 2014]. For example a recent study of 4^{th} grade science tests investigated the amount

of qualitative spatial reasoning tasks involved and demonstrated automated systems perform relatively well at these tasks [Forbus, 2016].

A brief introduction of widely used qualitative spatial representations is given below and we attempt to align them with applications from recent literature. The most widely used temporal qualitative relations are Allen's Interval Algebra (IA) [Allen, 1983], which express a unique temporal relation between any pair of intervals of time. We describe the specific qualitative representations of human observations used by our mobile robot in Chapter 4.

Topological Relationships

Topological relations describe qualitative relations between objects in space which are invariant under topological transformations. In topology, any continuous change to a space which can be continuously undone is allowed. It is common to take regions of space as the primitives rather than points. Often this involves obtaining MBRs, from an object-based abstraction method, which surround objects of particular interest. For representing and reasoning, two principal topological relations are the RCC (region connection calculus) [Randell et al., 1992, Cohn, 1996, Cohn et al., 1997] and the n-intersection model [Egenhofer and Franzosa, 1991].

Region Connection Calculus (RCC) is a calculus which uses regions of space as the primary spatial entity and represents topological relations between pairs of regions. It has been used to reason about MBRs extracted from 2D video data. It is based on a reflexive and symmetric primitive relationship between two spatial regions, a and b, defined as $\mathrm{C}(a, b)$. The intended topological interpretation of $\mathrm{C}(a, b)$ is that two regions a and b are connected if and only if their topological closures share a common point where the spatial regions are non-empty regular subsets of some topological space [Randell et al., 1992, Chen et al., 2015]. There are a number of relations defined by $\mathrm{C}(a, b)$, as shown in Table 2.1.

Relation $\mathrm{C}(a, b)$	Interpretation
$\mathrm{DC}(a, b)$	a is disconnected from b
$\mathrm{P}(a, b)$	a is a part of b
$\mathrm{PP}(a, b)$	a is a proper part of b
$\mathrm{EQ}(a, b)$	a equals b
$\mathrm{O}(a, b)$	a overlaps b
$\mathrm{PO}(a, b)$	a partially overlaps b
$\mathrm{DR}(a, b)$	a is discrete from b
$\mathrm{EC}(a, b)$	a is externally connected with b
$\mathrm{TPP}(a, b)$	a is a tangential proper part of b
$\mathrm{NTPP}(a, b)$	a is a non-tangential proper part of b
$\mathrm{Pi}(a, b)$	b is a part of a
$\mathrm{PPi}(a, b)$	b is a proper part of a
$\mathrm{TPPi}(a, b)$	b is a tangential proper part of a
$\mathrm{NTPPi}(a, b)$	b is a non-tangential proper part of a

Table 2.1: Table of relations defined on two regions of space a and b, by $\mathrm{C}(a, b)$ taken from [Randell et al., 1992].

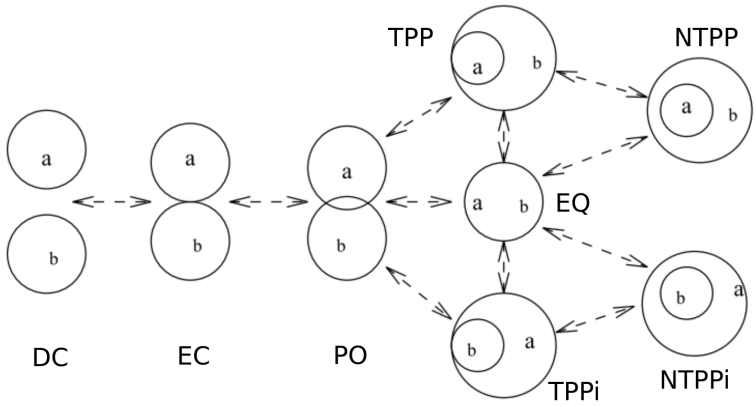


Figure 2.1: Eight JEPD relations shown between two regions a and b. This establishes the RCC8 language. The arrows show the next relation, assuming continuous movement or deformation. This is known as a conceptual neighbourhood [Fernyhough et al., 1998, Chen et al., 2015]

Eight JEPD relations defined by $C(a, b)$ that form the smallest set of base relations to allow topological distinctions is called the RCC8 language, namely: {DC, EC, PO, EQ, TPP, NTPP, TPPi, NTPPi}. These eight relations are shown as a conceptual neighbourhood diagram in Figure 2.1. If the boundaries of the spatial regions are not considered, which is often the case with MBRs, the RCC5 language is more suited and contains relations {DC, PO, EQ, PP, PPi}. In the computer vision domain, it is common to use a reduced (more coarse) set of relations to represent MBRs in the image plane, namely: DC (disconnected), PO (partially overlap), and P (part of) which is a compound of RCC5 relations {EQ, PP and PPi}: This forms the RCC3 language often used in the image plane [Tayyub et al., 2015].

There are many applications in the literature which use a form of RCC in order to abstract common or repeated patterns from quantitative 2D visual observations. Once an object-based abstraction of a video sequence is performed, common arrangements of objects can be learned using RCC relations, e.g. common table place settings for a meal [Dubba et al., 2010]; simple activities for daily living from a static camera dataset [Tayyub et al., 2015]; and even perform reasoning about spatio-temporal events being observed [Dubba et al., 2011].

Unsupervised learning approaches with RCC8 have also been used to represent and learn repeated activities in a multi-camera dataset consisting of 2D videos of aeroplane turnarounds [Sridhar et al., 2010]. The activities learned are deliberate movements which occur often between similar tracked vehicles in the dataset and are considered as activities of interest. The view point of the cameras is static and the objects are tracked and projected into a ground plane over the duration of each video. However, similarly to our work, the authors use an object-based visual and qualitative abstraction method to reduce the effect of slight visual variations between multiple observations. Each observation is represented as a histogram over spatial-temporal relations in order to learn common patterns.

An alternate topological framework, which is often used for spatial interpretation in Geographical Information Systems, is the n-intersection model [Egenhofer and Franzosa, 1991]. It is based on Point-Set Topology and uses the notion of *interior* and *boundary* of geometric objects; it considers regions of interest as sets of points embedded in a specified space, for example $\mathbb{R}^2$ or $\mathbb{R}^3$. Considering the simplest case, a region is a homogeneous 2D point-set a embedded in $\mathbb{R}^2$ and has the following three point-sets defined:

1. Interior, denoted (a°), is defined to be the union of all open sets that are contained in a.
2. Closure, denoted ($\bar{a}$), is defined to be the intersection of all closed sets that contain a.
3. Boundary, denoted (∂a), is the intersection of the closure of a and the closure of the compliment of a.

Considering just the interior (a°) and the boundary (∂a), the relationship between any two simply connected 2D regions a and b can be characterised by a 2x2 matrix called the four-intersection matrix shown in Figure 2.2 (left). Taking into account the closure ($\bar{a}$) point-set also, this is extended to the nine-intersection matrix show in Figure 2.2 (right). Further, it can be shown under certain assumptions about the nature of the regions involved that there are exactly eight valid matrices that correspond to RCC8.

$$\mathrm{R}(a,b) = \begin{bmatrix} a^\circ \cap b^\circ & a^\circ \cap \partial b \\ \partial a \cap b^\circ & \partial a \cap \partial b \end{bmatrix} \qquad \mathrm{R}(a,b) = \begin{bmatrix} a^\circ \cap b^\circ & a^\circ \cap \partial b & a^\circ \cap \bar{b} \\ \partial a \cap b^\circ & \partial a \cap \partial b & \partial a \cap \bar{b} \\ \bar{a} \cap b^\circ & \bar{a} \cap \partial b & \bar{a} \cap \bar{b} \end{bmatrix}$$

Figure 2.2: Four-intersection (left) and nine-intersection (right) matrices applied to two regions a and b [Egenhofer and Franzosa, 1991, Chen et al., 2015]

Directional Relationships

Directional relations can be used to describe where spatial entities are placed relative to one another in a qualitative manner. They usually consist of three primary elements: the target object, the reference object and a reference frame. But this can be reduced to two by taking an implicit reference frame. Two common binary point-based calculi in the literature are the cone-shaped direction and the projection-based direction shown left and right respectively in Figure 2.3 [Frank, 1996].

The basic relations of both the cone-shaped and projection-based calculus are obtained by partitioning up space. This is achieved by dividing a compass into four or eight disjoint sectors or cardinal directions. Cone-shaped relations are obtained by using angular directed lines going through the reference point, whereas projection-based relations are decided by the horizontal

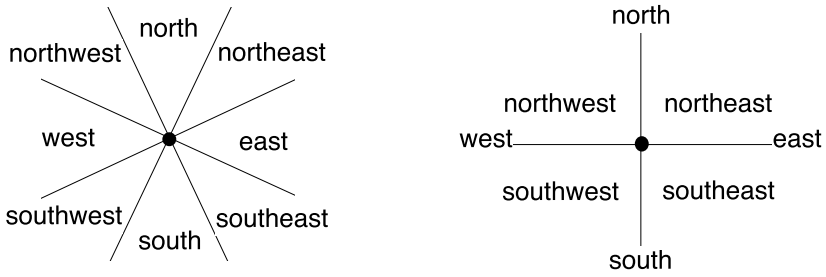


Figure 2.3: (left:) Cone-shaped directions. (right:) Projection based directions [Frank, 1996]

and vertical lines across the reference point. Both have the property that the area of acceptance for any given direction increases with distance [Frank, 1992, 1996].

Proposed in [Goyal and Egenhofer, 2001] is a Cardinal Direction Calculus (CDC) representation which uses direction relations between bounded connected plane regions. In this calculus the reference object is approximated by an MBR leaving the primary object unaltered. The CDC relations are then represented by 3x3 Boolean matrices. CDC has been used along with RCC relations to abstract object-object relations given visual scenes to learn a representation of objects movements [Ranasinghe and Karunananda, 2006]. Further, the direction of estimated human body joint movements have previously been used to learn human intentions for better human-robot interactions. The qualitative descriptors are scale and speed invariant and represent human actions as a histogram of direction vectors [Chrungoo et al., 2014].

Non-topological Relationships

Non-topological relations include distance, motion, size and shape among others. Spatial representation of distances can be categorised into those that represent absolute distance (between two objects) and those that represent relative distance (between objects relative to a third object). To use a distance representation to reason about space, directional information is also required. One approach is to combine distance metrics with a directional relationship to reason about distances and directions in geographic space [Frank, 1992]. Previous works have used a distance based qualitative abstraction, defined by the relative distance between human body joints and objects of interest (in particular the floor) [Zhang and Tian, 2012]. The authors detects abnormal activities in daily living activities such as elderly people falling; however this analysis is performed in a supervised setting, using an SVM and data collected from a static a RGBD camera. Another approach create a relative feature vector using distance and velocity between a person's hands and key objects in the scene for the purpose of learning, recognising and predicting egocentric manufacturing tasks [Behera et al., 2012b].

Qualitative Trajectory Calculus (QTC) is a popular non-topological motion calculus and one that we use in our work. It has multiple variants, but the idea is that once positional

information is determined by distance and direction relations, movement between two moving point objects can be qualitatively represented using functions, for example the following three functions define a simple variant known as QTC_{B22} [Van de Weghe et al., 2004, Delafontaine et al., 2011]:

1. movement of one object with respect to a second object's location;
2. movement of the second object with respect to the first object's location;
3. relative speed of the first object with respect to the second object.

Representing the values qualitatively, for functions 1 and 2 we use "−" to represent motion towards the second object, "+" to represent motion away and "0" to indicate an absence of relative motion. For function 3, we use {−, 0, +} to represent lower/ same/ greater speeds respectively.

QTC has been used to represent human dancing activities [Chavoshi et al., 2015]. The authors use a sophisticated infra-red motion-capture system to detect the exact position of each dancer's body which allows them to recover repetitive patterns which are post-associated with specific dance actions. Also robotic navigation can be enhanced by altering the robot's velocity based upon detecting humans in the environment and computing their QTC relations relative to the robot, i.e. human-aware navigation [Dondrup et al., 2014].

The effects of multiple different qualitative abstractions (RCC, QTC, direction, etc.) are evaluated in a robotic setting with the goal of improving an agents performance in a RoboCup soccer simulator [Young and Hawes, 2015]. A combination of directional, distance and size qualitative spatial representations have also been used to better understand static visual scenes observed by a mobile robot [Kunze et al., 2014] and for better object detection and discovery [Southey and Little, 2007, 2013]. Most of the works discussed in this section use a qualitative abstraction in order to generalise across multiple observations, of what is essentially the same activity, but which all may be slightly different in quantitative space. This facilitates the learning of common qualitative patterns, each for a different end-goal purpose. It is this ability to generalise quantitative sensor observations into more general relational models that we draw upon in our work.

Learning Spatial Relations

Qualitative spatial relationships can either be manually specified in advance, for example using one of the above calculi, or they can be discovered from observational data. The benefit of learning relationships automatically is that they are instantly relevant to the behaviour of the domain under observation, however a limitation is that all the data must be observed before any representation or learning can take place. The literature includes learning composite spatial-temporal relations between tracked regions which represent moving objects in real life domains, for example moving vehicles on a stretch of motorway [Fernyhough et al., 1998], or

similar moving-point objects represented as trajectories [Mohnhaupt and Neumann, 1991]. In [Fernyhough et al., 1998] an attention control mechanism is used to identify potentially interacting objects and primitive spatial relations such as *right*, *ahead* and *behind* are manually defined. Composite relations are then learned from observations, such as *following*, *pulling out* and *pulling in front*. These can relations can be associated with driving activities. Spatial models are derived from statistical evidence that "normal" behaviours are more frequently observed than "abnormal" behaviours. An activity model is then learned between interacting objects using statistically frequent data relating to the objects relative position and relative direction of motion. However, limitations of this work is that it is constrained to handle interactions between only two objects represented as MBRs, from a static frame of reference. Also, it is unable to learn composite relations representing more temporally extended behaviours, such as pulling out followed by overtake and pulling in front, which could describe a standard vehicle "overtake" behaviour.

Probabilities associated with learned composite activities can be explicitly computed, which has the advantage of facilitating the use of probabilistic models. Spatial relations, composite activities and their probabilities have all been learned and passed into variable memory length Markov models (VLMMs) to generate predictors of typical patterns of behaviour in a domain [Galata et al., 2002]. Non-topological relations have also been learned by creating a relative feature vector using distance and velocity between pairs of moving point objects [Behera et al., 2012a]. These feature vectors are then clustered to obtain component atomic events with which to describe human manufacturing-like activities from an egocentric vision set-up. This approach worked well but relies on a known and fixed set of objects where interactions are then recognised between them and wrist worn marker IDs. Spatial relations have similarly been learned for the purpose of improved object classification. A model of the general 3D spatial relationships between objects found in human environments is learned using a maximum entropy model of the underlying spatial regularities between the objects across environments [Southey and Little, 2007].

However, each of these approaches rely on analysing the observed data in an offline process. Relations are usually learned by taking an entire dataset of interactions between objects and learning suitable relations over the data. We manually define the qualitative representation in advance for our mobile robot, since the ultimate aim is to learn incrementally, and so a pre-processing stage is not appropriate. However, recently I co-authored a method that learns a qualitative representation incrementally, using natural language to guide the segmentation of various continuous feature spaces extracted from observations, whilst simultaneously using that representation to describe the observations [Alomari et al., 2017b].

2.2.2 Open Source QSRs

Qualitative representations are often considered largely theoretical tools to perform qualitative reasoning. However, the implementation of various toolboxes are making the integration into

real applications and software systems much easier and more common. One general attempt in that direction is SparQ[2], a qualitative spatial reasoning toolbox which aims to allows easy integration into applications and contains many QSR calculi [Wallgrün et al., 2006]. Another such toolbox is the Qualitative Algebra Toolkit (QAT)[3] [Condotta et al., 2006], and SPARQS, a software for the automatic derivation of composition tables [El-Geresy and Abdelmoty, 2004]. However, some of these systems compute qualitative representations from input point-like objects, their main focus is on symbolic reasoning, i.e. they assume that knowledge is already expressed in a symbolic form.

This thesis is accompanied by an open source qualitative spatial representation library known as *QSRLib*[4] [Gatsoulis et al., 2016a], that I co-authored. The goal of QSRLib is to allow for efficient and easy abstraction of quantitative sensor data into multiple qualitative representations and can be considered complementary to the above reasoning systems. The qualitative representations used throughout this thesis are computed using QSRLib under the Robotic Operating System (ROS) architecture[5]. Further, qualitative representations of time-series data, which is introduced in Chapter 4, is also implemented and available in QSRLib.

QSRLib has already been used in a wide range of robotic applications, including scene understanding by combining spatial relations with object class recognition [Kunze et al., 2014]; comparing qualitative and metric scene understanding [Thippur et al., 2015]; and for improving robot navigation by computing a qualitative relation to represent human motion in order to perform human-aware navigation [Dondrup et al., 2014].

2.3 Activity Analysis for Robotics

Advancements in the reliability of autonomous mobile robot platforms means they are well suited to continuously update their own knowledge of the world based upon their many observations and interactions. Activity recognition from mobile robots is a much more recent field of activity analysis research, in part due to the advancements in navigation, localisation and planning using probabilistic robotics techniques [Thrun et al., 2005]. This has allowed mobile robots to have much more accurate and reliable estimates of their own location within their map representation of the environment, and better able to perform actions based upon that estimate.

This was highlighted by a successful indoor office marathon by a PR2 robot platform [PR2 Robot Platform, 2017], in order to test the reliability of a navigation framework in a real-world office environment [Marder-Eppstein et al., 2010]. Long-term robust reliability was also the focus of the EU funded STRANDS robotic project[6], where multiple MetraLabs mobile

[2] http://sfbtr8.uni-bremen.de/project/r3/sparq/
[3] http://www.cril.univ-artois.fr/~saade/QAT/
[4] http://qsrlib.readthedocs.io
[5] http://ros.org
[6] strands-project.eu

robots [MetraLabs, 2017] were operational for a combined duration of 365 days autonomously travelling over 350km and performing 23,000 defined tasks in long-term installations in security and care environments over a period of four years. The early part of this project is summarised in [Hawes et al., 2016]. These capabilities have allowed mobile robots to co-exist for long periods of time in dynamic human-populated environments and allowed for a novel opportunity for human activity analysis on mobile robot platforms to learn from their own experiences.

A similar EU robotic project, RACE[7], focusses more on learning representations of observed behaviour in order to improve future robotic behaviour. For example in a restaurant domain where the robot could play the role of the waiter. The aim is to enhance the behaviour of an autonomous PR2 robot by learning from conceptualized experiences of previous performance, based on initial models of the domain and its own actions [Hertzberg et al., 2014]. However, much of the human activity learning was performed in a supervised setting, with little focus on long-term autonomy. Our aim is to learn similar representations of human activity from an autonomous mobile robot in an unsupervised learning domain.

2.3.1 2D Motion Patterns

Previous work using mobile robots to learn human motions represented by 2D trajectory patterns includes [Bennewitz et al., 2002, Cielniak et al., 2003, Bennewitz et al., 2005]. These works use multiple robots and a statistical approach to learn and classify human 2D trajectory motions using Expectation Maximisation (EM). However, in these works the mobile robots are positioned *statically* such that their sensors cover almost the complete region of interest resulting in fully observable trajectory paths. This requires prior knowledge of where interesting areas are and pre-defining the robot's positioning to obtain the complete trajectories. Furthermore, since they cover the entire region with laser scanners, they segment complete trajectories between common "resting points", which equates to pre-defining the start and end positions of the observed trajectories and therefore of potential motion patterns. An important feature also is that similar motions patterns have similar lengths, i.e. number of poses. This is in contrast to our work where a detected human trajectory has arbitrary length and the start can be detected at any point in a pre-built map of the environment, which might not be the actual source location of a motion behaviour of interest.

Similarly, the EM algorithm has been used to learn sequences of utterance time series data [Oates, 2002]; although this work is performed using an mobile robot with embedded sensors similar to ours, it uses utterances extracted from audio instead of visual features. More recently, an SVM has been used to learn trajectory features: speed, area covered, etc. using multiple robot mounted laser-scanners [Kanda et al., 2009]. However, their system is also heavily reliant on detecting complete trajectories via a large grid of laser sensors and requires pre-defining feature classes which is not possible in our unsupervised setting.

[7] http://project-race.eu/

Although these approaches use robots, they can be considered similar to a conventional surveillance setting, with the majority of approaches using the robot(s) as static sensors where the field of view does not change and the view point is usually carefully chosen so as to maximise information recorded. In this thesis we do not make the assumptions found in the work discussed above. We allow our single autonomous mobile robot to observe its operational environment capturing human detections which are abstracted into a qualitative space in order to generalise and learn common qualitative patterns of human motions. This helps alleviate some of the challenges faced with quantitative approaches.

Motion predictions of human trajectories is also commonly performed by automotive vehicles [Large et al., 2004], however the learned patterns of motion are commonly relative to the vehicle itself, where the aim it to avoid pedestrians and other road uses. This is similar to predicting the qualitative trajectory states of a detected human using QTC relations and adjusting the control of the mobile robot accordingly [Dondrup et al., 2014]. Similarly, autonomous pedestrian collision avoidance systems have been introduced [Llorca et al., 2011], but they do not perform the learning task on the car (or robotic) platform, and an offline processing stage is required.

2.3.2 3D Human Body Pose Activities

Recently, with the availability of cheap depth sensors and smaller GPUs fitted to mobile robots, the locations of an estimated human body pose is more easily extracted in real-time from onboard the robot. Learning a representation of whole body human activities from mobile robots has previously been performed, albeit in a strictly supervised setting.

Simple human body pose activities have been learned and recognised using the position and height of a person's detected face [Govindaraju and Veloso, 2005]. Further, the human body pose has been abstracted using qualitative 3D cone bins to create motion vectors (histograms) independent of ego-motion in order to represent and learn different actions [Xia et al., 2015]. Qualitative directions traversed by body joints have also been represented as a histogram to better learn human-robot interactions [Chrungoo et al., 2014]; and also body joint location covariance descriptors have been used for action recognition [Hussein et al., 2013]. A combination of quantitative body joint locations, qualitative hand positions, image HOG and motion features are used in a maximum-entropy Markov model in order to learn activities for daily living [Sung et al., 2012]. Each of these approaches attempt to create a compact and viewpoint invariant representation of the human body pose over a sequence of interesting frames, which is similar to our approach abstracting into a qualitative representation. However, the majority are performed on "clean" video datasets where each video sequence consists of a single activity instance, and where a fixed set of objects of interest is known in advance and extracted. Further, the robot is always ideally placed to record the interaction, and although cluttered scenes are often used, the activity rarely involves aspects of the environment and so the tracked person

is centre of the image and the environment can be ignored.

A related work in the literature uses a combination of descriptors on a visual dataset that was collected from a mobile robot patrolling a dynamic, human-populated environment, i.e. a university student area [Gori et al., 2015]. This is a very similar setting to the ones described and used throughout this thesis. However, a supervised SVM learning methodology is used to classify the different activity classes after each video instance has been temporally segmented and labelled with a ground truth activity. The key difference between this approach and our work is that our activity learning is performed in an unsupervised setting, with a focus on learning "in-the-wild" where no temporal segmentation is performed, and learning is incremental and lifelong. A further advantage is that our methodology encodes human-object qualitative relations as key features, whereas this work only uses human-robot features or visual features extracted from the human body pose. This allows our method to adapt to new and changing environments, as it selects the most frequently observed co-occurring human-object features in order to describe common human activities present in the environment.

Similar to the perspective of a mobile robot's limited field of view is recent literature that performs activity analysis from egocentric vision. Qualitative representations have been used in a system to assist with manual assembling-like tasks from an egocentric perspective [Behera et al., 2012b, Bleser et al., 2015]; to solve navigation and localization tasks for objects and robots [Wagner et al., 2004]; human robot interactions are recognised in [Xia et al., 2015] using a mixture of skeleton pose estimate features, optical flow and Space-Time Interest Point (STIP) features; early recognition of actions is performed in [Ryoo et al., 2015], where the idea is to learn a pattern not just of the activity itself, but of the short period of time before the activity begins in order to perform early prediction of such an activity in the future. However, each of these egocentric approaches use a supervised learning methodology. Unsupervised methods are discussed in detail in Section 2.4.

2.3.3 Learning Environment Representations

An important task for intelligent mobile robots is to understand their environment in a similar way as humans do. This will facilitate more useful interactions between humans and robots. For example, there is recent work teaching mobile robots to ground natural language commands that relate to specific objects, actions or spatial relations that occur in real-world scenarios [Boularias et al., 2015, Alomari et al., 2017b]. Further, imagine being able to call a robot to the "front of the room", when the room only has an explicit "front" due to objects in the room and their specific functionalities. In the literature, work understanding static scenes has used point-cloud information to reason about the stability and un-safeness of objects in a scene [Zheng et al., 2013]. Context-dependent spatial regions (CDSRs) are regions defined based on a combination of their functional use and their geometric properties. Further, there is research that enables a robot to discover, represent and reason about CDSRs [Hawes et al., 2012]; the cognitive

system first generates a qualitative spatial representation from a mobile robot's sensor data and represents the boundaries of CDSRs using anchor points. This is done in an unsupervised setting, and training CDSRs are transferred to a new situation using a structure-mapping comparison technique.

In this thesis, our mobile robot requires object discovery techniques to autonomously learn key object locations in its environment. This has previously been achieved using 2D images, that is, researchers have learned object models from visual SIFT features that are matched between sequential pairs of images to identify groups of moving features [Southey and Little, 2006]. However, our mobile robot takes multiple 3D scans of the environment in order to segment out object locations based upon convexity of objects, this is presented in [Bore et al., 2017, Alomari et al., 2017a].

2.4 Unsupervised Learning

For the purpose of lifelong learning of human activities from an autonomous mobile robot, a more task-appropriate learning setting is unsupervised learning, where each visual observation does not require offline manual annotations from an expert. For this reason, we draw comparisons between our learning task and that of an information retrieval task where probabilistic, generative methods such as Latent Dirichlet Allocation (LDA) [Blei et al., 2003] have been used in order to define a probabilistic model of a collection of discrete data, and draw inferences from. It was developed for extracting a set of interesting *topics* from a corpus of natural language text documents by using the co-occurrence of words between documents to learn document and topic level parameters. It extends previous work in Latent Semantic Indexing/Analysis (LSI/LSA) [Deerwester et al., 1990] and probabilistic LSA (pLSA) [Hofmann, 2001]. A review of similarities between LSA, pLSA, LDA, and others is presented in [Buntine and Jakulin, 2006].

Probabilistic generative approaches have been commonly used to learn about smart home domains, where sensor-based abstraction methods are usually used and patterns of co-occurring sensor readings are learned from occupants moving around their home. Examples include learning occupancy models from smart office environments [Castanedo et al., 2011]; learning person routines over long periods of time [Castanedo et al., 2014]; a temporal-LDA model is introduced for social media content created per user over time [Wang et al., 2012]; and learning trends in dynamic natural language topics that evolve over time [Bolelli et al., 2009].

2.4.1 Vision Based Systems

These unsupervised techniques have also been used to learn human activity categories from visual data, which is more similar to our domain. Here, often low-level image features are extracted using a pixel-based abstraction, such as, Space-Time Interest Point (STIP) features [Niebles et al., 2008]. The authors extract STIP features from the video sequences which

work well to concentrate on the moving human body across images, however they do not explicitly model the human body pose and categorise only somewhat simple human motions such as, *walk, wave, jump* or *spin*. An intermediate step in their method is to compute a code book of unique features, which is similar to our work. However, a major problem cited in their work is "The lack of spatial information provides little information about the human body, while the lack of longer term temporal information does not permit us to model more complex actions that are not constituted by simple repetitive patterns" [Niebles et al., 2008]. Descriptive spatial-temporal correlogram features, which encode the correlation of pairs of gray scale values in a co-occurrence matrix, have been used previously to attempt to address this issue [Savarese et al., 2008], however, their approach still suffers from low-level image processing frailties, and the requirement for a single person in the scene during a controlled training period. We address and partially alleviate this problem by using semantically meaningful qualitative features extracted from an interval graph representation of video sequences. Such features encode more "longer term temporal information" than used in the previous works. Further, our code book of unique features is adaptable and computed incrementally to accommodate different environments or activities the robot might observe. Each code word contains semantic qualitative relations with key landmarks in the environment which helps understand more complex activity interactions.

Similar works include using pLSA combined with local shape context descriptors on silhouette images [Zhang and Gong, 2010]; a combination of semantic and structural features to learn actions, faces and hand gestures [Wong et al., 2007]; and finally, good results were obtained in learning human action categories by fusing a vocabulary of local spatio-temporal volumes (cuboids) with a vocabulary of spin-images to capture the shape deformation of the actor [Liu et al., 2008]; Each of these approaches use an unsupervised learning framework similar to the ones presented in this thesis, however the data is often recorded using a static camera and are mostly restricted to a single person, face or moving hand, clearly defined in the centre of the image. Further, each video sequence usually consists of a single activity instance temporally segmented, that is, the actions have been performed without the variability of a mobile robot's frame of reference, and none have attempted to learn human activities from video data not previously segmented into action sequences. The visual observations our mobile robot records can contain multiple people interacting in the scene simultaneously, where the person or the activity observed can be incomplete, noisy or interrupted. We move towards an "in-the-wild" setting, where video sequences are not manually temporally segmented and learning can be performed incrementally.

Scene Understanding

Unsupervised techniques have also been used in the object classification and scene understanding literature, for example, pLSA has been used to find the location of objects in images for the purpose of better understanding visual scenes [Sivic et al., 2005]. Scene understanding

methods also use pLSA in [Quelhas et al., 2007], where *bags-of-visterms* are used to represent extracted histograms of quantized local visual features from images. 3D object structure can be learned from images in an unsupervised setting by taking multiple images and generating a 3D image model [Rezende et al., 2016]. A Hierarchical Dirichlet Process mixture model has been used with low-level optical flow features to learn more complex interactions between groups of moving pedestrians [Wang et al., 2009]. However, each of these unsupervised works does not extend to learning human activities over a temporal sequence of images.

2.4.2 Qualitative Based Systems

There is relatively little focus on the application of unsupervised learning of qualitative spatio-temporal relationships to understand human activity from a mobile robot platform, which is the focus of this research. One related approach in this space, albeit not of human activities, used a qualitative abstraction and a statistical approach based on frequencies in order to learn common patterns of moving cars from a fixed camera set-up. A second has used an unsupervised Markov Chain Monte Carlo (MCMC) approach, coupled with a qualitative representation in order to generalise and learn patterns across multiple video sequences of aeroplane turnaround videos [Sridhar et al., 2010, Sridhar, 2010]. Here, a qualitative spatial calculus abstracts near continuous video using RCC relations between multiple tracked vehicles. We draw similarities to this work due to its graph-based representation of observed QSRs between pairwise objects. The graph, known as an *Activity Graph*, is searched over in order to obtain a *graph covering* that best describes the video sequence in terms of activities (sub-graphs) taking place, i.e. discovering activities and their instances in the videos. However their video sequences are somewhat more simple, they are recorded from multiple static cameras and large moving object locations are detected and tracked in an offline setting. Activities taking place consist of a small number of objects interacting with respect to the camera frame of reference. This is unlike human body pose activities which contain much more variation, and there is an unknown number of objects of interest in an environment.

Although we use a similar qualitative abstraction of visual observations and graph representation, we do not attempt to fit a graph covering to an Activity Graph. This method relies on encoding the entire video dataset and using an offline MCMC technique to obtain the most appropriate sub-graphs that relate to activities. The learning methods we use for human activities are updated on the robot, incrementally improving performance with new observations over time which is more efficient than batch sampling methods. These are a crucial aspect of an autonomous robotic lifelong learning system.

2.4.3 Interpreting Unsupervised Models

One particular challenge when using unsupervised learning techniques are that the learned patterns can be difficult to interpret. It is also difficult to visualise topics as they are often

very high-dimensional multinomial distributions over a vocabulary of code words. Some novel methods have been developed to handle this in recent literature. An exemplar-based interpretation to learned high-dimensional distributions over large corpora proposed using low-rank approximations and matrix decomposition [Chen et al., 2009]. However, the common approach to understanding a *topic* distribution is to obtain the set of natural language words that have high probability in each topic distribution, and present these to a user. Given such a set, it is often easy to conceive the meaning of a specific topic or corpus. This technique is used often in the literature [Chang et al., 2009], since an end-user has a lot of prior semantic knowledge about natural language words and the relations/similarity between them. However in our work, our code words are not natural language words, but graph structures of qualitative relations. Whilst these code words contain more semantic information than similar pixel-based features, e.g. STIP features, there is no defined similarity distance between them, meaning this technique is less informative to understand coherent topic distributions.

Specifically for LDA, tools for the exploration of learned topic models have been developed. For example *LDAvis* with open source implementation[8] introduces the idea of *relevance* of each code word. It provides a user interface where a topic of interest can be selected to visualise the top-probable code words, or a code word can be selected to identify each topic where it has a high probability of occurring [Sievert and Shirley, 2014]. Similar to the notion of relevance, [Bischof and Airoldi, 2012] introduce Hierarchical Poisson Convolution (HPC), a model to rank code words for a given topic in terms of both the frequency of the word under that topic as well as the words exclusivity to the topic. Another visualisation tool allows a user to navigate, using drop down boxes, through a corpus by selecting topics and linking them to highly probable code words and corresponding documents in a learned topic model [Chaney and Blei, 2012]. Further, it has been demonstrated when understanding topics derived from micro-blogging sites such as twitter, that aggregated messages can help improve classification performance [Hong and Davison, 2010]. Topic distributions have also been analysed using a novel *intrusion* experiment, where a topic distribution (a set of highly probable code words) have other random words manually added and the idea is for a human reviewer to pick out the new code word that does not match the topic, i.e. the intruded code word [Chang et al., 2009].

2.5 Summary

This chapter has introduced and reviewed several of the main research areas relevant to unsupervised human activity analysis and mobile robotics. Some of these areas are expanded upon further in the remaining thesis, others are introduced just for reference. A key idea in this thesis is to take a qualitative-based abstraction method of visual sensor data observed by a mobile robot. We described alternate sensor and low-level pixel-based methods which are unsuitable for our task due to the variability of the robot's observations. Secondly, commonly

[8] https://pypi.python.org/pypi/pyLDAvis

occurring patterns in the data are learned in an unsupervised setting. The most similar applications that make use of unsupervised learning methods to solve problems in recent literature have been introduced. Finally, we propose practical solutions to challenging problems when applying this framework on a real-world deployed mobile robot "in-the-wild", which have not been fully addressed in recent literature.

Chapter 3

Human Activity Observations

Our aim is for an autonomous mobile robot to understand human activities from long term observation of human populated environments. In this chapter we first define a human activity from the perspective of a mobile robot. We focus on the perception data available to the robot via its multiple sensors; that is, how and what the robot observes, detects and encodes within its limited field of view. Human observations are encoded using a leg-detector, an upper-body image classifier, and a convolutional neural network for body pose estimation, while objects within the environment can be automatically segmented from a 3-dimensional point cloud representation, if they are not known in advance. This process results in a continuous stream of quantitative data available to the robot. In the later chapters, we show how the robot uses this data to obtain a conceptual model of human activities taking place in its environment.

This chapter comprises of three main sections. First we define what we consider as a human activity and the specific activity domains the robot is required to operate in. Secondly, the robot is introduced, its sensors and how it interprets the environment. Lastly, we describe how the robot encodes human observations using one of two techniques, 1) *human trajectories*, or 2) as more detailed *human body pose* sequences.

3.1 Human Activities

We introduced the term *activity* to relate to a temporally dynamic configuration of some *agents*, where the agents can be grounded in the real world, or could be online agents, etc.. In this research we aim to 1) understand human activities as motion patterns performed in real human environments, and 2) for that system to scale to allow continual learning in complex dynamic environments. We focus only on single human activities. To do this we explore the interaction between the human agent and environment, namely between a human and key objects which provide functionalities [Agre and Chapman, 1987]. We therefore define a *human activity* to be a temporally dynamic configuration of a human agent relative to close-by *key objects* in the

environment. We make the following assumptions and definitions related to human activities:

- A *key object* is a semantic entity with a fixed location in an environment which provides some functionality that may be required for the execution of certain activities of interest in that environment [Kirsh, 1995].
- A *human activity* is considered as a partially ordered sequence of sub-activities (or repeated patterns) between positions of a person's body joints relative to key objects. In turn, these patterns (or sub-sequences) can be thought of as one or more simple qualitative relations holding between a person's body joints and/or a number of objects in the environment. For example, a person "picking up a cup" might relate to a specific human activity comprising of the sequence: "reaching", "grasping" and "lifting" performed by the person's hand with respect to the cup.
- A *human motion behaviour* is considered as a simple human activity, based upon the evolution of the 2-dimensional position of a person with respect to surrounding key objects. That is, when the sequence of observations does not include detailed body pose information.

These assumptions are common in the literature [Lavee et al., 2009, Hamid et al., 2009, Aggarwal and Ryoo, 2011]. The definition of a human motion behaviour and human activity are expanded upon in the next section by introducing the concepts of *activity granularity* and *abstract class*.

The resolution of human activities we are interested in is somewhat limited by the available perception or sensory inputs to the robot. This manuscript provides a framework for a mobile robot, and therefore the perception is limited by its sensors and field of view capabilities. As an example of this, imagine you live in a blocks world, and repeatedly see a pattern of two moving blocks. It would not be too difficult to understand an activity with respect to these moving blocks. However, if you have more perceptual details about the blocks, e.g. their colour, you might be able to learn a different granularity of pattern, e.g. the blue block moves before the red block, but other times the red block moves first. This could result in learning two patterns of activity, even if they are the same pattern when colour is not taken into consideration. This is a key limitation to our system; since the performance of state-of-the-art robot perception is still far from human level perception. This translates as effecting the robot's ability to detect objects (static or moving) within its environment, and similarly to the blocks example, it can only learn activity patterns at a particular level of granularity.

The level of abstraction of possible learned activities is also restricted by the order in which observational data is made available to the robot. For example, if you always perceive two blocks moving at the same time as each other, you could believe that it is a repeating pattern of activity. However, if you first see each block moving individually and then observe those same blocks moving together, you might believe there are two different patterns and that

the latter instance involved both occurring simultaneously. In this manuscript, we show that the robot can learn repeated patterns of human activity from a mobile robot's observations. However this is limited to the sequences of data the robot actually observes and is liable to similar mistakes as in this blocks example.

A second key concept of human activities is the notion of an *abstract class*. The human brain reasons with the environment by abstracting away much of the detail. For example, a specific mobile phone is one instance of our concept of all mobile phones, and you would expect it to be able to make phone calls like other mobile phones, since this is the main function of all mobile phones. This is an example of an object's abstract class. One mobile phone is a particular *object* in the *class* of all mobile phones. Knowledge about object classes is important to a robot and can be learned over time by observing multiple object instances that make up a specific class [Sridhar et al., 2008].

This concept extends to dynamic human activities. For example, one person "making coffee" is a specific activity *instance* in the *class* of all people making coffee, even though the exact spatial and temporal quantitative details of how each person performs the activity differ, just like the exact details of all mobile phones differ. It is the goal of the robot to learn the crucial qualitative details of different activity classes by observing multiple activity instances in order to recognise an activity in this class in the future.

We make the assumption that human activities performed with slight variability in the spatial quantitative space are considered members of the same activity class. This means that activities carried out in a visually similar manner, using similar key objects in the environment, can be considered as instances of the same activity class. One limitation is that semantically similar activities that are not visually similar are considered as different activity classes. For example, under our assumption, two instances of a "making coffee" activity would be considered different activity classes if they appear visually different; e.g. one instance is performed by boiling a kettle and pouring water and coffee into a mug, and the second is performed by putting a mug in a coffee machine where the machine prepares the drink itself. These two activities are comprised of different qualitative patterns between the human and different key objects in the environment. At a subject level of analysis, one might learn that the two coffee making classes both proceed a "drinking" activity, and hence are semantically similar.

Activity classes highlight the importance of both object involvement and object class within an activity class. For example, given two instances of a human activity, one where object A is used and a second uses object B, these could be considered as different activity class. However, if object A and object B belong to the same object class, e.g. a mug and a cup, it is likely that the two human activities belong to the same activity class. We do not propose new methods for learning object classes in this work.

To summarise, in this research we focus on learning single-human activities, where the level of activity granularity and abstraction are not manually assigned in advance. Similarly, activity classes are not manually defined in advance and in fact, these are learned by the robot from multiple observations of human activity instances. However, the robot is restricted by its sensor limitations and the current state-of-the-art computer vision techniques. Given these limitations, the robot extracts qualitative descriptors from encoded observations, a process which is introduced in Chapter 4. These qualitative descriptors are considered as sub-activities in which partially ordered sequences are learned and considered to be *human activity classes*.

3.2 Robot & Environment

In this section, we introduce our mobile robot, our representation of the human environment and the visual observations available to the robot. The robot's sensing modalities present a continuous stream of quantitative data which allow the robot to build an understanding of the world it inhabits. In the next sections we describe how the robot encodes human observations using state-of-the-art methods and how it segments key objects in its environment.

3.2.1 Mobile Robot

As part of the EU STRANDS project, each project partner has a Metralabs Scitos A5 mobile robot [MetraLabs, 2017], four of which are shown in Figure 3.1. This is the robotic platform used for the work described in this manuscript. We will give specific details about this platform, its sensor modalities and how it perceives the world. Note that the techniques presented in this manuscript are hardware independent and modular.

Each STRANDS Metralabs robot is capable of performing a variety of tasks over long periods of time, such as long-term navigation and autonomous docking. A single robot is used during long term robotic deployments. The reader is pointed to [Hawes et al., 2016] for a

Figure 3.1: STRANDS Metralabs Scitos A5 mobile robots. From left to right: BoB, LUCIE, Linda and Betty. Identifiable are the head mounted ASUS RGBD cameras atop a pan-tilt unit, and touch screen display.

detailed account of a two month deployment set in a real world office environment. During the deployment, the robot maintains a *routine* which outlines the allowed daily working hours and also the weekends/public holidays when it is required to remain stationary. The *scheduler* then accepts units of work defined as *tasks*, based upon the specified requirements for the day. This thesis focusses on a single robot learning human activities from multiple observations, and hence, a single mobile robot is used to detect and track humans as they pass within the field of view of its sensors. However, all the STRANDS robots are equipped with the following array of sensors:

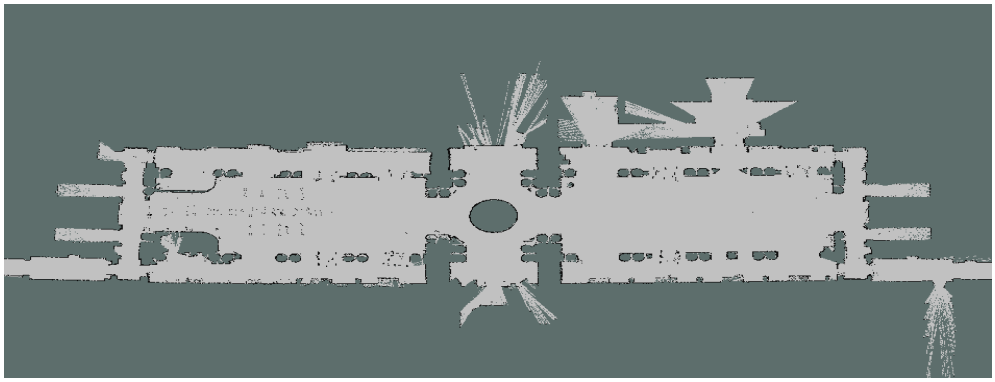
- a base mounted (SICK S300) laser range finder for mapping and localization. Maximum range of 30m and horizontal operating angle of 220°.
- Two ASUS Xtion Pro-Live RGBD cameras. One chest mounted for the purpose of obstacle avoidance, and the other head mounted atop a pan-tilt unit which is used to detect people in the environment. Each with an operating range of 0.5m to 3.5m and operating angle of 58° horizontal, 45° vertical and 70° diagonal.

The robots also have a touch screen display, and three on-board PCs each running ROS (Robot Operating System) Indigo [Quigley et al., 2009] and the full STRANDS software system [Spatio-Temporal Representations and Activities for Cognitive Control in Long-Term Scenarios STRANDS project, 2017].

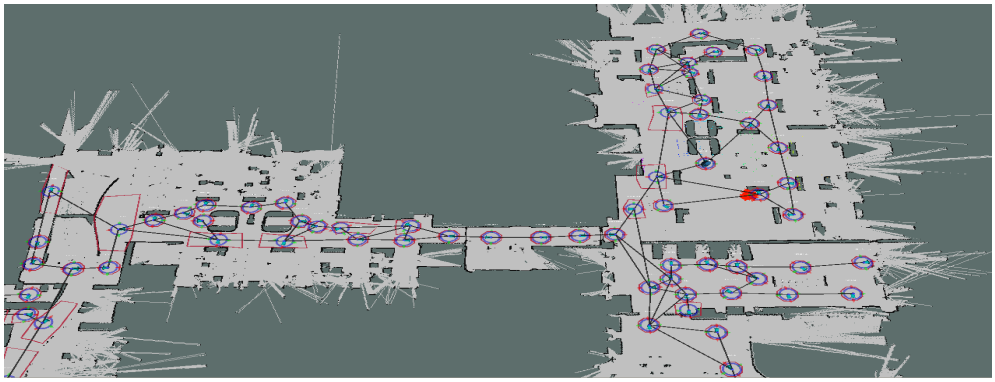
3.2.2 Map Representations

The mobile robot uses an occupancy grid to represent the 2-dimensional environment as a binary map. The occupancy grid is referred to as the robot's *metric map*. The metric map allows the robot to understand the layout or floor plan of an environment, i.e. which parts of the environment contain free (or occupied) space, and also where the robot is located within its environment. The occupancy grid is generated using the off-the-shelf ROS package gmapping [ROS gmapping, 2017], which provides a ROS wrapper for OpenSlam's Gmapping as described in [Grisetti et al., 2007]. It uses a laser-based SLAM (Simultaneous Localization and Mapping) Rao-Blackwellized particle filter to create a 2D occupancy grid map from laser and position data collected by the robot. An example occupancy grid map is shown as an image in Figure 3.2a, where black cells represent occupied space such as walls and furniture; light-grey cells represent free space and dark-grey cells are outside the explored region. One point in the occupancy grid is considered the origin and assigned $(0, 0)$ coordinates, and canonical axes are chosen so that each cell has an (x, y) coordinate. Artefacts arise when the laser briefly passes through a window or a doorway which is not fully explored, such as in the bottom right corner of the image.

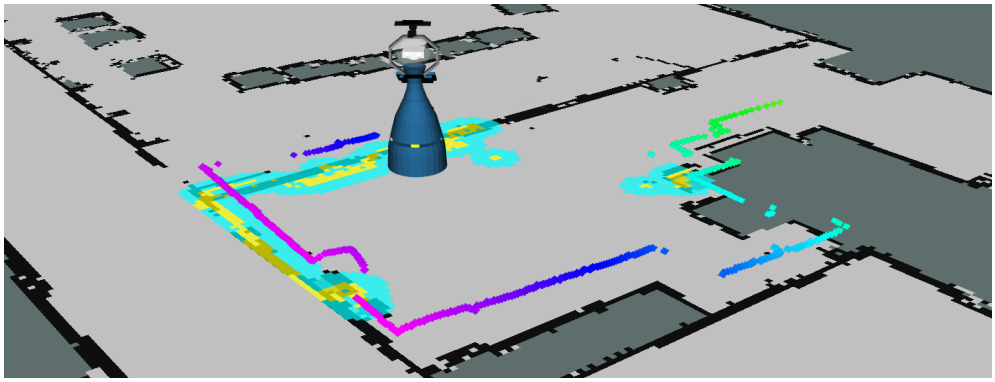
The metric map is a low-level representation of whether space in the 2D environment is free or occupied and each cell corresponds to an (x, y) coordinate. On top of this we build a semantic layer to allow the robot to reason about the environment with a higher-level semantic



(a) Metric map. Occupancy grid generated with gmapping ROS package. Black cells represent occupied space; light-grey cells are unoccupied space and dark-grey cells are outside the explored region.



(b) Topological map: connected nodes (known as waypoints) and edges overlaid onto a metric map. Each waypoint has an influence zone, marked in dark-red around the corresponding.



(c) Local costmap (light blue) overlaid on to the metric map, with current laser scanner update (purple-dark to blue-green). The robot icon represents the robot's estimation of its position in the map.

Figure 3.2: Map representations. Best viewed in colour.

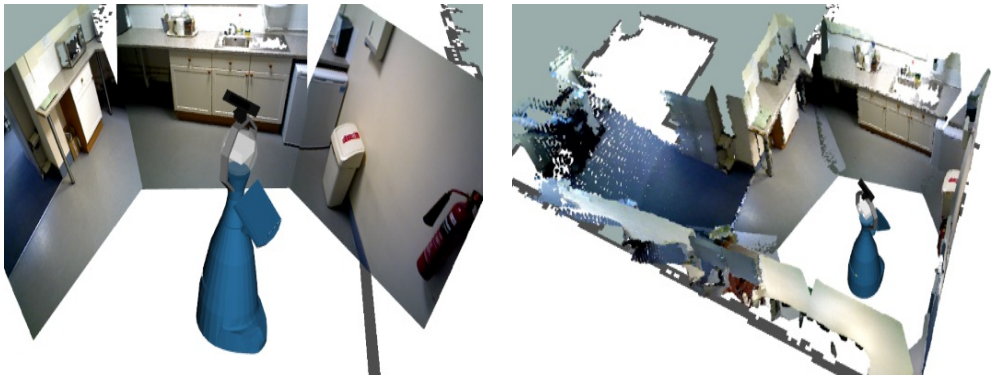
understanding. One such semantic layer is known as the *topological map*. A topological map consists of a number of abstract locations represented as nodes (referred to as *waypoints*), the surrounding influence zone, and connecting directed arcs (known as *edges*) labelled with the movement behaviour. The robot uses this map for topological navigation, which is described as moving between waypoints (nodes) using the specific motion behaviour specified on the labelled edge (arc). Further, the robot uses these pre-specified abstract locations as interesting positions in the environment where tasks can be performed. An example topological map can be seen overlaid onto the metric map in Figure 3.2b. This map is manually pre-built before a long-term deployment, however there are tools and ROS services to extend it autonomously in the STRANDS navigation stack.

A 2-dimensional region directly surrounding the robot in the map is defined as the *local costmap*. The occupancy of this space is updated using the laser-based SLAM implementation, while simultaneously keeping track of the robot's estimated location within the map (similar to generating the metric map). The local costmap is used when generating short-term navigation plans and gives the robot a robust localised position within its environment, especially when mobile. An example of this process can be seen in Figure 3.2c, where the robot constantly updates its local costmap (light blue) using the data received from its laser scanner (pink-blue-green). This allows it to efficiently keep track of the space immediately surrounding it, and move safely even in the presence of dynamic obstacles.

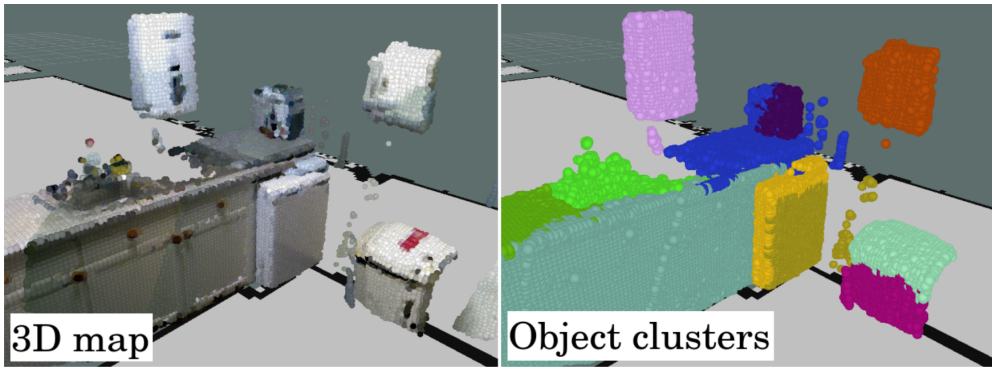
3.2.3 Object Representation

A second key component in the robot's environment are *objects*. In this thesis we focus on objects which people interact with in daily living and which provide some functionality for human activities. For example, a person might walk up to a printer-copier machine, stop in front of the machine in order to perform an action (swipe a key card to log in) and whilst doing so they spatially interact with the object. For this reason our representation of human activity includes relative positions of people with respect to key objects within the robot's environment. However, detecting and tracking arbitrary objects in real time from a robotic platform is a very difficult and an unsolved problem. Therefore to learn the position of interesting objects within an environment, the robot first pre-builds a 3D model of its environment by fusing together multiple RGBD images. The three-stage process can be seen in Figure 3.3a: 1) the robot moves its pan-tilt multiple times capturing an RGB image and a corresponding depth point cloud for each angular position. This process is known as a *sweep*; 2) it registers each pixel in the depth point cloud with an RGB value from the corresponding RGB image; 3) multiple registered point clouds are then fused together to create a large point cloud representation of the robot's entire environment (covered by the sweeps).

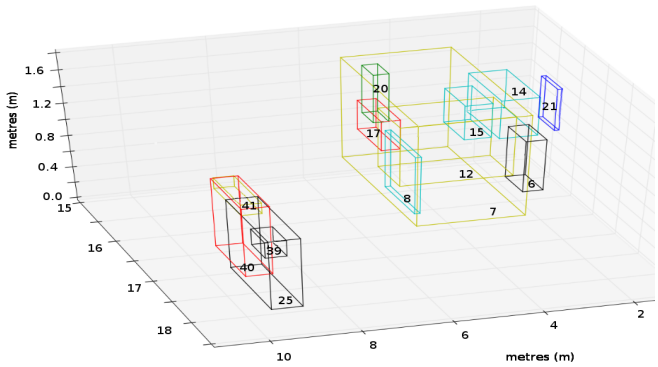
Once the robot has its 3D point cloud representation of its environment, it can extract locations of potential objects by rendering the surface using *surfels* (surface elements) [Pfister et al., 2000] and extracting clusters of pixels. From the fused point clouds, state-of-the-art per-



(a) Generating a 3D representation of the environment. (left) The robot moves its pan-tilt multiple times capturing an RGB image and a point cloud at each angle. (right) The robot fuses together the registered point clouds to create a single 3D representation.



(b) Segmenting candidate object locations. (left): Surfel representation of the robot's 3D environment. (right) Candidate object locations automatically segmented as clusters.



(c) Subset of candidate key objects aligning to the highest scoring locations in the environment where people stop and interact.

Figure 3.3: Object representations. Best viewed in colour.

formance in extracting semantically meaningful segments can be achieved using an unsupervised segmentation algorithm which is grounded in the convexity of common human objects. This is demonstrated in [Schoeler et al., 2015], and we use a method similar to that presented in [Bore et al., 2017]. It first splits the point cloud into a collection of *supervoxels* [Papon et al., 2013] over which an adjacency graph is formed. Then, weights are assigned to the edges in the graph based on local convexity of the point cloud and colour difference between different segments. Finally, to segment the point cloud, iterative graph cuts are performed to separate parts with concave boundaries and/or large colour differences. An example of the surfel representation of the 3D environment can be seen in Figure 3.3b (left) and the resulting clustered segments can be seen (right). We consider these segments as candidate key objects within the robot's environment.

Unlike a standard supervised method, the unsupervised method presented here allows the robot to segment any (potentially new) environment it may encounter, i.e. it is not restricted to previously learned environments, a set of supervised object models, or to a dataset of standard object classes. This is an advantage of this technique which motivates unsupervised human activity analysis as how people interact with these key object locations. However, there is an existing problem in the literature relating to unsupervised techniques for understanding which objects are interesting given a particular environment or task, and what level of abstraction is of particular interest. For example, a supervised object detector can recognise objects from its training dataset, however, unsupervised techniques have no pre-specified dataset and therefore the level of abstraction is not specified in advance, e.g. given an observation of a fridge in a kitchen, it is unclear to the robot which part is of particular interesting: is it the fridge handle for a grasping task, or the door, since this part moves. This is subjective, based upon defining a task. One way this problem has been approached in the literature is by combining visual features (which are used to predict an object's abstract class) with a semantic web hierarchy [Young et al., 2016, 2017]. Higher abstraction levels in the hierarchy are then used to automatically label the objects of interest into classes which maintain sufficient information.

To concentrate the robot's attention on only objects that are part of observed human activities, the trajectories of humans in 3D space are analysed to extract the locations where people frequently stop and interact. The candidate key objects (extracted above) are scored according to their proximity and interaction frequency with regards to people's hands. The highest scoring objects are considered as "interesting" in the environment. Figure 3.3c shows the extracted positions of the high scoring candidate object segments based upon the university kitchen environment scans shown in Figure 3.3a and a collection of human observations introduced in the next section. On the right-hand side of the environment, object IDs 6 and 21 in Figure 3.3c are seen to relate to the segmented trash bin and paper towel dispenser respectively which are clearly visible in the scans. These segmented object clusters correspond highly to locations where people stop and interact. The corresponding segments of the 3D point cloud are extracted and defined as key objects with respect to human activities.

Examples of such extracted segment clusters, which align well with real and useful objects pertaining to activities, can be seen in Figure 3.4.

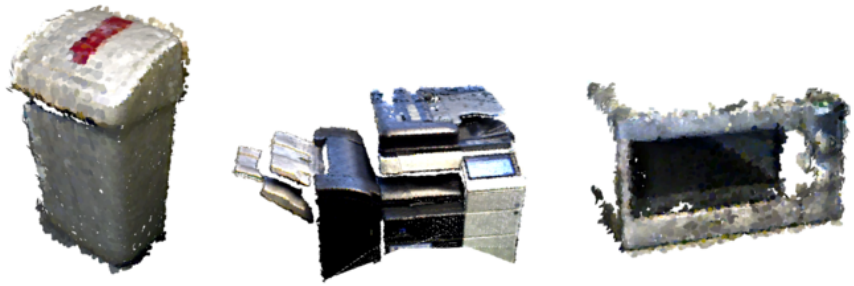


Figure 3.4: Segments extracted from the 3D representation and registered with corresponding RGB data. (left:) Trash bin. (centre:) printer/copier machine. (right:) microwave.

Given a set of key objects in the robot's environment, it is possible to visualise their locations in a Semantic Object Map known as *SOMa* [Kunze and Hawes, 2017]. SOMa consists of two layers, which can be used separately depending upon the task:

1. Multiple (x, y) coordinate points on the metric map are connected to form a semantic Region of Interest (ROI). These are often used to segment out semantic regions of space, e.g. different rooms or office boundaries. These can be seen as yellow and blue connected polygons in Figure 3.5, where three ROIs, student room (light blue), kitchen (yellow), and staff room (blue) have been manually specified, however there are techniques to learn interesting regions autonomously in the STRANDS software stack.

2. The position of key objects can be visualised in SOMa as (x, y, z) coordinate points in the robot's 3-dimensional map frame of reference, where the metric map lies at $z = 0$. An example of visualising static object positions is also shown in Figure 3.5 where brightly coloured CAD (Blender) models are overlaid onto the metric map.

To conclude this section regarding object representations, the robot is able to take multiple point cloud sweeps from its pan-tilt mounted RGBD sensor, fuse them together and segment out a set of candidate object clusters. The candidate objects are then ranked by whether people interact with them or not, and a set of key objects in the robot's environment are learned. Both semantic regions and semantic object locations can be represented and visualised using a SOMa map. It is worth noting that our representations and learning framework discussed in the next chapters would extend trivially to include dynamic objects detected in real-time by the robot.

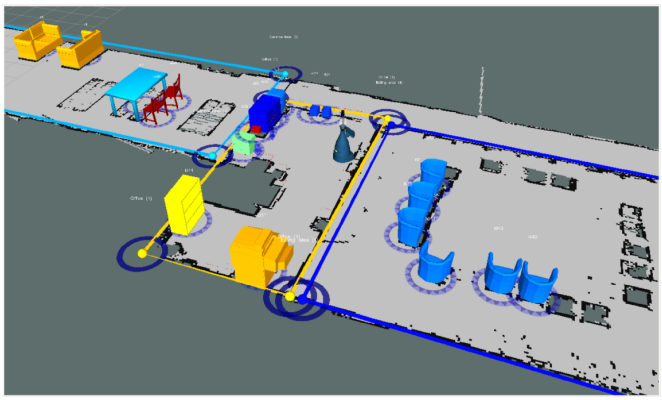


Figure 3.5: SOMa: Semantic object map showing multiple Region of Interests (yellow/blue polygons) and SOMa objects as brightly coloured CAD Blender models overlaid onto the robot's metric map.

3.3 Human Detections

In this section we describe the human observations made by the mobile robot. We describe how the robot uses its on-board sensors to detect and encode the behaviour of humans who pass within the sensor's field of view. This process produces a stream of noisy and incomplete quantitative observations of humans observed within the robot's environments. For example, a person might walk straight past the robot, being in its field of view for only a couple of seconds. Alternatively, multiple people could be within the sensor's field of view for a long time, however, they could easily occlude each other causing a more difficult detection problem.

The robot uses two types of representation for human behaviour it detects and observes. In the following sections we first define a *human trajectory* and later a more detailed *human pose sequence*. These quantitative observations are abstracted into a qualitative representation which is introduced in the next chapter.

3.3.1 Human Trajectory

The robot has a fixed, base mounted laser scanner which is used to produce a metric and local cost map, i.e. to understand its static and dynamic 2D environment. To obtain an estimated position of a person within the laser's field of view the 2-dimensional range scans are constantly input into an off-the-shelf human leg-detector introduced in [Arras et al., 2007] and implemented as described in [Bellotto and Hu, 2009]. This gives the robot an estimated position of a person relative to the map and invariant to visual noise which often affects camera systems, e.g. lighting or motion variabilities. A secondary detection method is also implemented; the robot uses its head mounted RGBD camera to classify images that contain a person using an upper-body detector described in [Mitzel and Leibe, 2012] using RGB images. This gives the robot an estimate of a person's position relative to the robot's head mounted camera's field of

view.

The robot's detections are often noisy due to a multitude of reasons:

- The sensor modalities used to detect people only grant a very limited field of view, i.e. they do not observe the entire environment at all times.
- The sensors can easily be occluded by (static or dynamic) obstacles in the environment, e.g. a person standing in a direct line in front of a second person, or a piece of furniture in the robot's field of view.
- The accuracy of the robot's localisation relative to its metric map is affected when the robot is mobile. Therefore the human detections represented relative to the map frame are subject to the same uncertainty.

An example image showing the real-time detections of multiple people using the combination of sensors is given in Figure 3.6. The image shows the position of the robot using a blue Blender robot model (central) overlaid onto the metric map. The robot accurately detects five people using the leg detector (each shown on the left as a person model overlaid on the metric map), and only three upper body bounding boxes of positions of people extracted from the head mounted camera's RGB image. Note the wider field of view of the laser scanner, compared with the ASUS RGBD camera, which allows it to correctly detect all the people in this situation. The robot uses the Bayes Tracker as described in [Dondrup et al., 2015] to stitch together the multi-sensor detections into a sequence of positions based upon their chronology. We consider a sequence of detections belonging to the same person as forming a human *trajectory*.

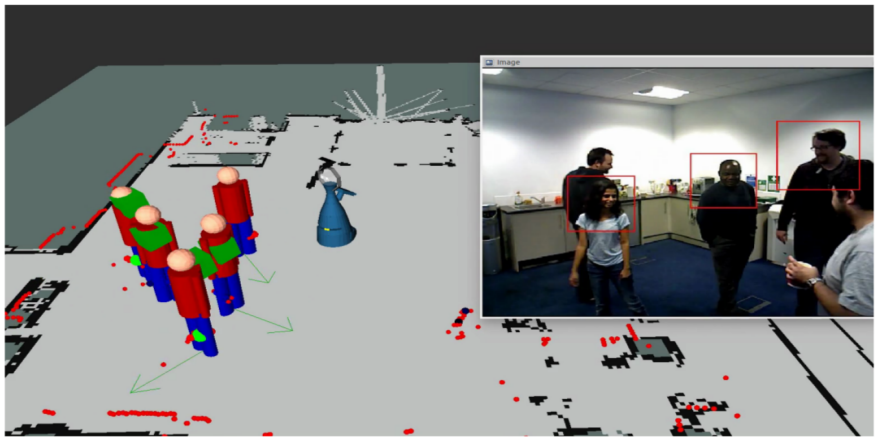


Figure 3.6: People detections from an implemented leg-detector, based on a base mounted laser scanner, and upper body detector using RGB images from the robot's head mounted camera. (Image taken from [Dondrup et al., 2015]).

Formally, we define a single *trajectory pose* as an (x, y) Cartesian coordinate in the map coordinate frame. For a detected person, we obtain a sequence of trajectory poses over a time

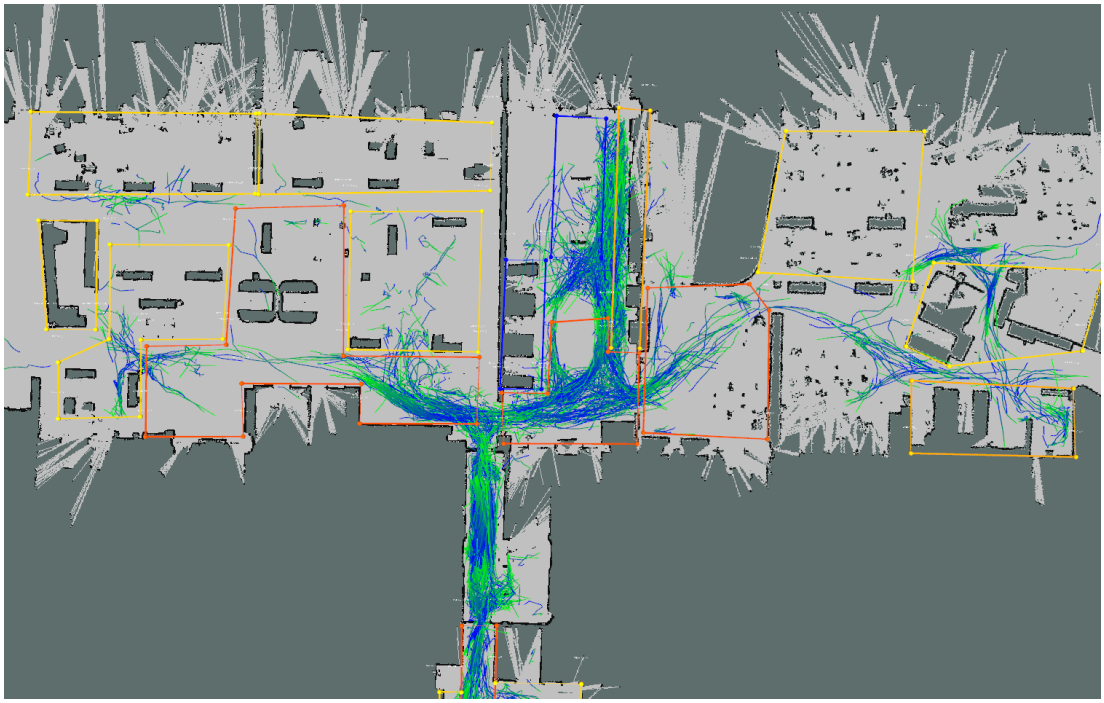


Figure 3.7: A collection of observed trajectories from our mobile robot whilst patrolling and monitoring an office-like environment (overlaid onto a metric map). Direction of motion of each trajectory is shown by colouring the trajectory poses blue to green. SOMa ROIs can also be seen as yellow and orange polygons. Best viewed in colour.

series of detections, and generate a *trajectory* based upon the chronology of these poses. A trajectory is defined as $T = [t_1, t_2, \ldots, t_i, \ldots]$, where each t_i is the detected trajectory pose at timepoint i. Note that there are no restrictions placed on i, i.e. each trajectory comprises of an arbitrary number of detections, based upon how long the person is detected by the robot. This variation is a major difficulty when using real world data to learn about human motions and behaviours on a mobile robot. Figure 3.7 shows a metric map overlaid with a collection of observed trajectories. The direction of motion of the observed person is shown by colouring the trajectory poses from blue to (increasing amounts of) green. It should be highlighted that the trajectories obtained using these techniques are always incomplete and often noisy. They represent only a section of a person's complete motion through an environment as observed by the robot during some period of time. Further, the accuracy of the detected poses is affected if the robot is even slightly mislocalised, which occurs frequently whilst the robot is moving.

3.3.2 Human Body Pose

Observing human trajectories is particularly useful to allow the robot to analyse and learn patterns of people's 2D movements and motion behaviours within the environment. However,

these learned behaviours represent only a portion of human activities being performed due to the detection of the person being abstracted into a 2D trajectory pose, i.e a single (x, y) coordinate relative to the metric map. For a more detailed understanding of human activities, the robot estimates the full human body pose from RGBD images observed from its head mounted camera. A *human body pose* comprises of a number of body joint locations, which loosely relate to body parts: head, neck, torso, shoulders, elbows, hands, hips, knees and feet. These joint locations are estimated in 3D Cartesian coordinate space, as opposed to a trajectory pose which is a single (x, y) point on the 2D map plane. Information about the body joint positions allows the robot to understand and learn more detailed actions and human activities.

A popular approach to obtain human pose estimates is to use the OpenNI person tracker [OpenNI organization, 2016] which uses only the sensor's depth stream to detect multiple persons and infer their 3D pose in real-time. An example grey-scale depth image can be seen in Figure 3.8 with the OpenNI human pose estimate overlaid where each body joint location is depicted by a blue circle with purple connecting lines forming the rough shape of a human skeleton. This approach works well on the robot for detecting people in real-time because it efficiently runs using the robot's depth image only, as opposed to more memory intensive 3-dimensional point cloud data.

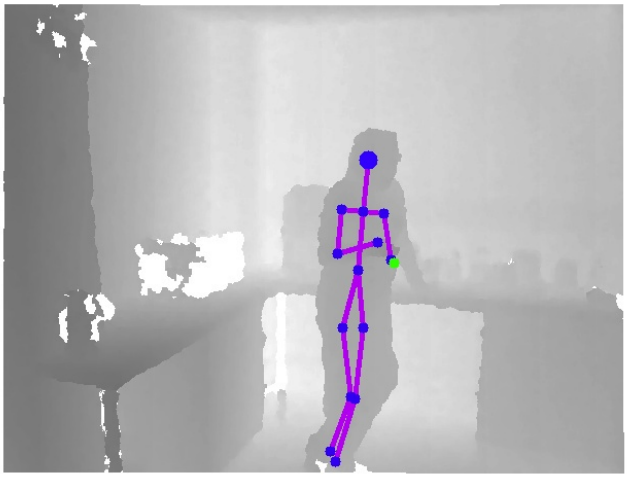


Figure 3.8: Real time OpenNi human pose estimate overlaid onto a grey-scale depth image. The 14 body joint locations are shown as blue circles. The right hand is shown as a green circle to distinguish it from the left. The person is backward facing.

For our robot to understand more complex human activities, it is especially important to obtain reliable body pose estimates in difficult cases of fast moving, human-object interactions from challenging viewpoints. Unfortunately, interactions between a human and an object cause problems for the OpenNI pose estimation, i.e. when there is not enough information in the depth image to distinguish between the two, the object is often inadvertently considered part

of the person's body. Two examples of where OpenNI fails to correctly infer the human body pose can be seen in Figure 3.9 (a) where the pose estimates are overlaid onto the RGB image as brightly-coloured ovals connecting body joint locations. OpenNI struggles with images with the following features:

- When a body joint location is occluded by another body joint, e.g. the right arm in left-most image is not visible by the depth sensor because it is occluded by the person's torso. This happens often when a person is side-on or at an angle to the sensor which is common given non-optimal view points.
- When the observed person is backward facing the orientation of the person is often incorrectly inferred.
- When a body joint is interacting or in contact with part of the environment or an object, e.g. the left arm in Figure 3.9 (a)(right).

In each of these challenging situations, the depth image does not contain enough information to correctly infer where 3D body joint location and hence it is not able to accurately estimate the human body pose. To mitigate these problems, we leverage RGB colour data to help distinguish between object and body joints and help resolve backward facing poses. Our pose estimation system operates in a two phase approach, firstly, the efficiency of OpenNI is utilized to detect people in the observable environment (in real-time on the robot's CPU). Secondly, the corresponding RGB image is fed as input into a state-of-the-art convolutional neural network "pose machine" (CPM) [Wei et al., 2016] to better estimate the 2D body pose (on a midrange GPU). Subsequently, we take the improved (x, y) coordinates in the camera frame of body joint positions from the CPM, and the corrected depth coordinate (z) from the original OpenNI depth image. An example of this approach can be seen in Figure 3.9 (b), which clearly shows the improved CPM pose estimates. Note the estimated arm and hand locations are greatly improved, even accurately detecting the reach of the hand joint in the right most image.

Formally, we define a *joint pose*, j, as an (x, y, z) Cartesian coordinate corresponding to a single inferred body joint location in the camera coordinate frame along with the corresponding (x_m, y_m, z_m) position translated into the map coordinate frame, i.e. $j = (id, x, y, z, x_m, y_m, z_m)$. The camera frame coordinates are equivalent to the blue circles in Figure 3.8, however the map frame coordinates rely on a transformation into the map frame using the robot's estimated location within the map. The accuracy of this transformation is improved by restricting the robot to recording humans only when static, however the uncertainty remains in the joint pose estimates. A human *body pose estimate* is defined as a collection of joint poses, one for each body joint of the person tracked, i.e, $p = [j_1, j_2, \ldots j_n]$, where $n = 15$ using the OpenNI/CPM tracker described above. Similar to encoding the trajectory of a person, for each detected person we obtain a sequence of body pose estimates over a time series

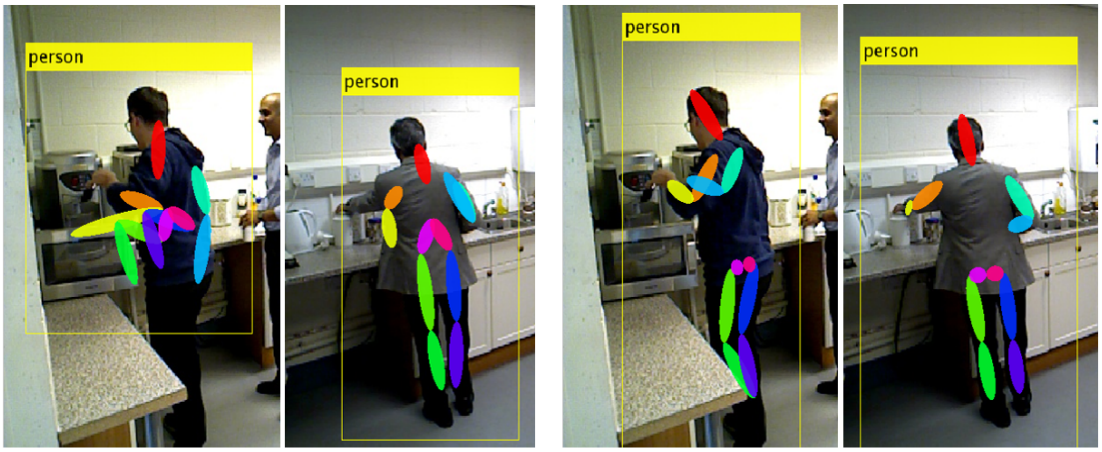


Figure 3.9: Comparison between (a) OpenNI body pose estimates using depth images only. (b) Improved body pose estimates using convolutional neural network "pose machine" on RGB images and mapping the (x, y) coordinate onto the depth image to obtain the corrected depth coordinate (z). Best viewed in colour.

of frames and generate a human *body pose sequence*. This is defined as $S = [p_1, p_2, \ldots, p_i, \ldots]$, where each p_i is the detected human body pose estimate at timepoint i. Again, there are no restrictions placed on i, i.e. each human pose sequence comprises of an arbitrary number of frames and therefore human body pose estimates.

3.4 Concluding Remarks

To conclude this chapter, our mobile robot is able to infer the 2D and 3D location of people (and their body joint locations) within its field of view using a combination of laser scanner, RGB and depth image techniques. It is able to translate these detections into the map coordinate frame and couple them with nearby objects of interest found using a state-of-the-art automatic segmentation of point cloud sweeps. An example of a human detection represented in the map frame of reference along with automatically segmented key objects of interest can be seen in Figure 3.10. Here the human body pose estimate is represented as a person model in the visualisation overlaid onto the metric map, with pink cubes to represent the hand joint positions. The registered point cloud data representing the segmented key objects is also overlaid, along with a robot to depict its pose and field of view.

The stream of quantitative observations presented in this chapter are stored on the robot using custom ROS messages which are described in Appendix A. In the next chapter, we show how the encoded observations are mapped into an abstract qualitative space in order to generalise patterns invariant to exact quantitative positions within the real environment. We then describe

Figure 3.10: An example human body pose observation relative to the environment. (left:) RGB image corresponding to a single human body pose detection. (right:) The human body pose estimate is translated into the map coordinate frame using the localised position of the robot and overlaid as a person model where the two hand joint locations are shown as pink squares. Also overlaid are the learned key objects using the registered point cloud segments.

how the robot auto-generates a vocabulary of discrete descriptors by extracting sequences of partially ordered qualitative relations from the encoded observations. This vocabulary is used in the later chapters to encode a vector space representation of the observations, facilitating the use of text analysis techniques to recover a set of classes which we consider as human activities.

Chapter 4

Activity Representation using Qualitative Relations

In this chapter, we introduce how the robot spatially and temporally abstracts the stream of quantitative observations obtained from the techniques presented in the previous chapter. To motivate this, consider that quantitative approaches provide precise, yet estimated, numerical information about the state of some entity. However this precise information is often inaccurate, unnecessary and is not usually available to humans. For instance, it is natural to omit specific numerical descriptions when describing a static environment or dynamic observation. Further, it is likely that a qualitative representation will suffice when describing a relationship between multiple entities not based upon a global frame of reference, e.g. 'the coffee is in the cup', or 'the cup is next to the sink'. This is usually sufficient information with respect to a given task, e.g. describing a "drinking coffee" activity. For this reason, our robot uses multiple different qualitative calculi to abstract away different aspects of observed human activity with respect to nearby objects, i.e. it uses relative distance, motion and angles which occur between encoded observations and key objects to differentiate different qualitative patterns.

This chapter is composed of two main sections. First, the observed human body pose sequences are abstracted into a *qualitative spatio-temporal representation*. The qualitative calculi used are described in detail in Section 4.2 and we discuss the motivation for using this representation. The second section of this chapter introduces our novel discrete descriptors which are extracted from the qualitative representation as sequences of temporally connected qualitative relations. This novel process is introduced in Section 4.4. For this purpose, we define an *interval representation* and an *interval graph*, which facilitate the extraction of the descriptors from the encoded observation. Given multiple observations over a period of time, the robot represents each as a bag-of-qualitative-words which allows the robot to employ Information Retrieval techniques in order to recover a set of emergent activity classes described in the next chapter.

4.1 Qualitative Motivation

To abstract quantitative human observations into a qualitative representation we use a *qualitative spatial representation* (QSR). We briefly discuss the main advantages of using a QSR before describing how each calculus is applied to the robot's observational data and the specific benefit of each to the application of unsupervised human activity analysis.

4.1.1 Dimensionality Reduction

Qualitative representations have often been implemented for the purpose of abstracting data to obtain a low dimensional representation whilst maintaining necessary (or discriminatory) information present in the encoded data [Chen et al., 2015]. They have been used for many practical applications in the literature, for example, to build a qualitative egocentric representation for navigation [Wagner et al., 2004], to learn by qualitative observation Young and Hawes [2015] or to learn qualitative pattern of specific tracked objects camera images [Sridhar et al., 2010]. Data abstraction is the main reason why the robot employs the use of a qualitative spatio-temporal representation of its observational data. By abstracting away the quantitative details, a QSR provides the robot's learning framework a low dimensional representation of both the sequences of observed people (input data), and the learned motion patterns and activity classes (output data). For example, the robot's observations consist of multiple quantitative poses per timepoint; one for each object, human body joint, or trajectory pose observed. These quantitative locations are embedded in a continuous feature space that can be thought of as the robot's entire environment. Over the lifetime of the mobile robot, this continuous space becomes intractably large and it becomes difficult to maintain or learn patterns over such a high dimensional space using the quantitative location values.

The robot abstracts the observed quantitative locations into a discrete, qualitative feature space where each point is mapped into one or more symbolic relations, e.g. one point might be considered as 'moving towards' or 'moving away' from another, or 'far away from' as opposed to 'close to'. These abstractions represent a much lower dimensional feature space, based upon the number of qualitative calculi used and the specific number of encoded locations. We will show in the later chapters that with careful choice of QSR calculi, discriminatory information about human motion behaviours and activities can be maintained. This abstraction also allows us to throw away unnecessary quantitative data, especially with a robot's limited on-board memory/computational power.

4.1.2 Generalising Observations

A second reason for using a qualitative framework is that we would like the robot to be able to generalise and draw comparisons across multiple observations. To do this, we assume that the exact temporal details or spatial locations of observations are not as important as the relative positions between observed entities. For example, if a person raises their hand above

their head and waves, the exact (x, y, z) coordinates of their hand or head are not particularly important; it is the relative movement which captures a possible "waving" activity. If the robot observes a qualitatively similar pattern, it can learn a representation of the common activity without the observations occurring in the exact same location in space or moving their arm in exactly the same fashion. Understanding a low dimensional qualitative pattern of the motion to represent "wave" can be more useful than trying to match the exact coordinates of the person's head/hands. This approach allows the robot to learn by generalising across multiple observations invariant to exact temporal durations or spatial locations.

Likewise, human activities can occur using different body joints, such as a person waving using either their left or right hand. As in some literature, the exact body joint is often abstracted to leave only the abstracted type of body joint when encoding qualitative relations; hence a human activity can be learned or recognised from observations when either hand has been used (where "hand" is the body joint type). This has the effect of reducing the dimensionality of the qualitative space further.

4.1.3 Partial Observations

One assumption in this work is that the mobile robot cannot observe its entire environment simultaneously using only its on-board sensors; which seems reasonable in most domains. Also, the robot cannot possibly observe a human's complete movement within the world and will usually be confined to a single room or building. This leads to two issues pertaining to unsupervised human activity analysis:

1. The first relates to the robot's particular viewpoints within its environment, given that it must select somewhere to "look". Depending upon the sensors' field of view at any point in time, the robot can obtain large variations in the quantitative data observed.

2. Secondly, the robot is usually restricted to a confined environment unlike the person, i.e. the person can leave the building and go home for instance. The robot can not observe the person's movements outside of its environment.

We address the first of these points in two ways, first the mobile robot can randomly select its own viewpoints from the set of learned key object locations in order to best observe humans interacting with them. This allows the robot to mainly observe useful human-object interactions in the environment. Secondly, we abstract the robot's observations into a qualitative representation which encodes a symbolic representation invariant to specific viewpoints. For example, if a person walking towards a printer is observed by a well positioned robot (so it observes the complete interaction between the person and object which we would consider a "good" viewpoint), it can encode a long trajectory or human body pose sequence with many quantitative poses. Conversely, the robot could have a "bad" viewpoint of the interaction and not observe it completely, i.e. it would generate a smaller number of quantitative poses due to its limited observation. In both cases the person performs the same interaction, yet the

quantitative encoded data varies greatly, the only difference is the location and viewpoint of the robot. Given that the robot will never be able to observe its entire environment at the same time, it encodes its observations using viewpoint invariant qualitative representations in order to learn patterns within these observations, irrespective of the robot's location.

The second issue of partial observations relates to the assumption that the mobile robot has a restricted environment due to some physical constrains within its domain, i.e. it cannot follow a person everywhere they might possibly go. This relates to only observing a small fraction of a person's total movements within the world. For example, before a human enters the robot's limited environment (maybe restricted by room or building), the robot has no information about the person and cannot possibly observe them. It also does not know or understand the person's intentions or goals. This could be addressed using a complex fixed camera set-up, where a person is tracked between all possible rooms or locations, however this is inefficient and not likely to capture a persons complete movements, e.g. if they go outside, or travel. We therefore consider the robot as always only partially observing a person's movements, resulting in observations which are never "complete". However, the aim is to observe enough human behaviour and activities to learn patterns using invariant qualitative representations.

Using a qualitative representation to abstract the robot's observations makes the framework somewhat invariant to varying length of observed pose sequences, noisy or incomplete observations. For example, if the robot observes people always walking towards and through a specific door, the robot can learn a pattern of behaviour that once the door is interacted with in a certain way, e.g. "move towards" and "touch", it is likely a person will walk through it, based upon its many qualitatively similar observations encoded invariant to exact viewpoint. Further, the robot can learn this behaviour from observing any doors in its environment by abstracting the object type.

4.1.4 Temporal Abstraction

Our qualitative framework also allows the robot to abstract observations in time. For example, human activities often occur over very different durations of time. That is, there are large variabilities between different human activity classes (*intra-class*) and also between multiple instances observed within the same activity class (*inter-class*). Inter-class differences are simple to imagine, that is, different types of human activity occur for different durations of time. One example is the activity of "standing still", where body pose estimates are easy to estimate, and can occupy thousands of frames; whereas a more complex task such as "opening a fridge" can take less than a second and can contain noisy body pose estimates due to body part occlusions and fast moving entities (objects and body parts). Intra-class differences are also common in human activities; there is large temporal differences between multiple instances of the same activity class. For example, two instances of a person "making a phone call" can occupy very different durations of time depending upon the person performing the task, the time constraints on the call, or the recipient of the call. However, the two activities may be performed in the

exact same manner with respect to spatial movements and object interactions.

In order to learn human activity classes invariant to time, we employ qualitative temporal calculi, which encode logical semantic relations based upon the temporally relative occurrence of encoded spatial movements, i.e. how two spatial movements occur relative to each other in time, abstracting away the exact temporal details, such as the number of frames the movements occurred for or their exact duration apart.

4.1.5 Summary

In summary, the large spatial and temporal variation of human movements coupled with the limitations of the robot's sensors present major difficulties when using real world data from a mobile robot, and motivates the use of invariant qualitative spatial and temporal representations. Abstracting the quantitative data into a low dimensional qualitative space helps to alleviate some difficulties and allows the robot to generalise multiple observations, draw comparisons and extract patterns. Examples of the multiple qualitative spatial calculi used are given in the next section, and we introduce the qualitative temporal representations in Section 4.3.2.

4.2 Qualitative Spatial Representations

As the robot observes people passing within its field of view, it encodes the quantitative detections using one of the previously introduced techniques as either a human trajectory (on the 2D map plane) or a human body pose sequence (of a collection of 3D body joint locations). The robot then encodes these observations using a qualitative spatial representation to abstract away the exact quantitative details of the specific instance. The robot achieves this by encoding the observation using the following three qualitative calculi:

- **Qualitative Trajectory Calculus** (QTC) represents the relative motion of moving point objects (MPOs) with respect to a reference line connecting them, and is computed over consecutive timepoints or frames [Van de Weghe, 2004, Van de Weghe et al., 2005a, 2006].
- **Qualitative Distance Calculus** (QDC) expresses the qualitative Euclidean distance between two points depending on pre-defined distance thresholds [Clementini et al., 1997, Weld and De Kleer, 2013].
- **Ternary Point Configuration Calculus** (TPCC) qualitatively describes the spatial arrangement of a point relative to two others in a 2-dimensional configuration, i.e. it describes the *referent*'s position relative to the line created by connecting two other points (the origin and the relatum) [Moratz and Ragni, 2008].

These three qualitative calculi each encode a different aspect of the quantitative observation, i.e. relative motion, distance or spatial arrangement. Next we give more details about each

of the chosen representations and how they are implemented based upon the observed human trajectory and human pose sequence observations.

4.2.1 Qualitative Trajectory Calculus (QTC)

Qualitative Trajectory Calculus (QTC) was developed as a qualitative calculus to represent and reason about moving point objects (MPOs) in free space. In our work, the trajectory poses and body joint poses are considered as MPOs and assumed to be disjoint and evolve continuously in free Euclidean space and time. QTC was first introduced in [Van de Weghe, 2004, Van de Weghe et al., 2004] and since then, multiple variants have been added, namely Basic type (QTC_B), Double-Cross type (QTC_C) [Van de Weghe et al., 2005a, 2006] (based on the Double-Cross Calculus [Zimmermann and Freksa, 1996]), Network type (QTC_N) [Bogaert, 2008] and Shape type (QTC_S) [Van de Weghe et al., 2005b]. Further, implementation constraints have been applied based upon discrete time [Delafontaine et al., 2011], as the MPOs are represented on a computer architecture, with finite sampling rate sensors.

In this work, we use the basic, discrete, variant QTC_B as described in [Dondrup et al., 2014] and implemented in the Qualitative Library *QSRLib* [Gatsoulis et al., 2016a,b]. This variant is equivalent to the QTC_{B11} subtype presented in [Delafontaine et al., 2011], where the added syntax represents the number of spatial dimensions considered. It defines the following three qualitative spatial relations between two objects o_1, o_2:

- o_1 is moving towards o_2 (represented by the symbol $-$);
- o_1 is moving away from o_2 ($+$);
- o_1 is neither moving towards or away from o_2 (0).

The possible relational states can be seen in Figure 4.1, represented as a *conceptual neighbourhood graph*, which defines transitions (arcs) between possible states (nodes) [Fernyhough et al., 1998]. Within each node of the graph, a solid dot represents a stationary object and an open dot represents a moving object with possible direction of movement denoted by a semi-circle. Finally, the symbolic tuple representation is given at the top of each node. For example, the top-left node in the neighbourhood diagram shows two open dots representing two moving objects (let's call the left open dot o_1 and the right dot o_2). Both o_1 and o_2 are moving to the right (represented by the semi-circles attached to the right hand side of the dots), and the qualitative tuple $(-, +)$ is shown at the top of the node since o_1 is moving towards o_2 (represented by the symbol $-$), and o_2 is moving away from the position of o_1 (represented by $+$).

Since QTC represents relative motion between two moving point objects in a qualitative manner, we consider it appropriate to encode certain aspects of a person's movements relative to their environment and represent it symbolically. The robot segments the environment into a set of key static objects (as described in Section 3.2.3), so the person's movements are considered with respect to these key locations and we can consider a subset of QTC_B relations, i.e. where

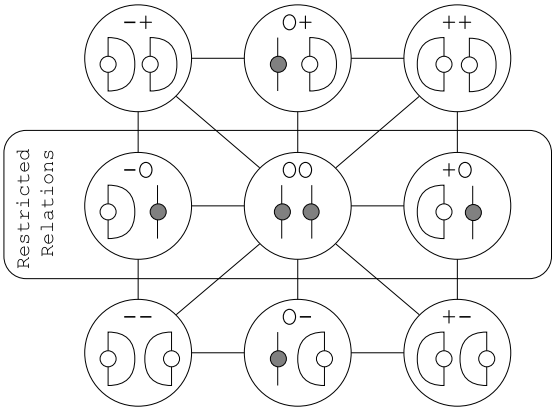


Figure 4.1: Relations of basic Qualitative Trajectory Calculus (QTC_B) represented as a conceptual neighbourhood diagram. Nodes are possible states, connected by arcs representing possible transitions. Within each state, solid dots represent stationary objects and open dots represent moving objects. Central row is highlighted as the represents the reduced conceptual neighbourhood diagram when one MPO is static.

one of the two MPOs is always static. For example, the QTC_B symbolic relation between a person's hand and a static object, such as a fridge, will contain at least one "0". This restricts the possible states (nodes) in the conceptual neighbourhood graph and reduces it to only the middle row (highlighted), with three possible symbolic relations $\{(-,0),(0,0),(+,0)\}$.

4.2.2 Qualitative Distance Calculus (QDC)

Qualitative Distance Calculus (QDC) abstracts the absolute Euclidean distance between two static points in 2D or 3D space. It depends upon pre-defined region boundaries and encodes a symbolic qualitative representation of the distance value. Formalised in [Clementini et al., 1997], along with various other qualitative representations of spatial and positional information, the following three axioms are defined for the concept of distance between two points in a quantitative space:

1. $dist(P_1, P_1) = 0$ (reflexivity),
2. $dist(P_1, P_2) = dist(P_2, P_1)$ (symmetry),
3. $dist(P_1, P_2) + dist(P_2, P_3) \geq dist(P_1, P_3)$ (triangle inequality),

where, in 3-dimensional Euclidean space distance between two points $P_i = (x_{i,1}, x_{i,2}, x_{i,3})$ is calculated as:

$$dist(P_1, P_2) = (\sum_{j=1}^{j=3} |x_{1,j} - x_{2,j}|^2)^{1/2}. \tag{4.1}$$

QDC represents the Euclidean distance between two points using a sequence of threshold boundaries, Δ, which must exhibit monotonicity, i.e. each boundary threshold, δ_i, is larger

than the previous one, $\Delta = [\delta_1 < \delta_2 < ... < \delta_i]$. Distances that fall between the thresholds are often given arbitrary natural language names for convenience. For example, the distance, d, between two points, P_1 and P_2, (i.e. $d = dist(P_1, P_2)$) can be represented symbolically, e.g. if $d < \delta_1$ the distance can be considered as "near", or $\delta_2 < d < \delta_3$ considered "far".

The intuition behind using QDC is based on the assumption that human motions can be partially explained using distance relative to key landmarks. That is, a set of QDC relations localises a person or body joint location with respect to a reference landmark or set of landmarks, and a change in the QDC relation can help explain relative motion of the person. The robot uses the learned key object locations as the landmarks in this intuition. Then the relative distance between an observed human trajectory pose or the multiple human body joint poses are computed and represented symbolically with respect to nearby key objects. Figure 4.2 shows an example QDC system applied to a key object segmented as an object cluster representing a trash bin (shown at the centre). The thresholds monotonically increase from the object, i.e. $\Delta = [\delta_1, \delta_2, \delta_3, \delta_4]$. Note the distances greater than the largest QDC threshold are often called "Ignore" which defines a region at a distance greater than $max(\delta_i)\ \forall i \in \Delta$ away from an object.

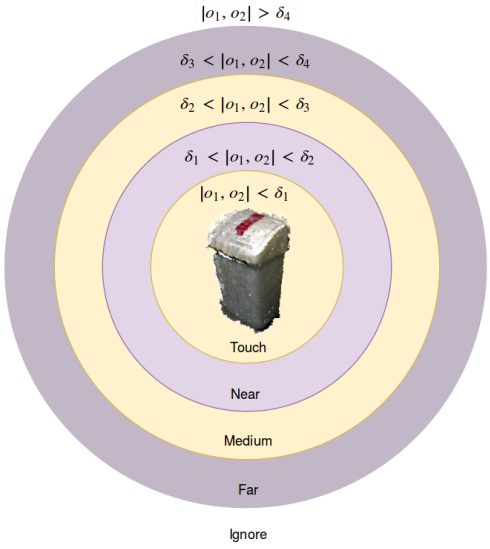


Figure 4.2: QDC system. Qualitative distance threshold boundaries ($\Delta = [\delta_1, \delta_2, \delta_3, \delta_4]$) applied in 2D Euclidean space to a key object location automatically segmented from the robot's environment.

4.2.3 Ternary Point Configuration Calculus (TPCC)

Ternary Point Configuration Calculus (TPCC) also deals with static point-like object locations, and is defined on the 2D image plane. It qualitatively describes the spatial arrangement

of an entity relative to two others [Moratz and Ragni, 2008]. Specifically, it describes the *referent*'s position relative to the *relatum* and *origin* positions, which together is known as a relative reference system where the origin is known as the anchor point. The full TPCC system is shown in Figure 4.3 (left), where the symbolic relations consist of triples of the following three sub-categories:

- {front (f), back (b)}: the referent can either be in "front" (f) or to the "back" (b) of the 2D line created by connecting the origin and the relatum locations.
- {left (l), right (r), straight (s)}: the referent is at an angle of either "left" (l), "right" (r) or "straight" (r) with respect to the 2D origin-relatum line.
- {distant (d), close (c)}: the referent can either be "distant" (d) or "close" (c) to the relatum with respect to the origin.

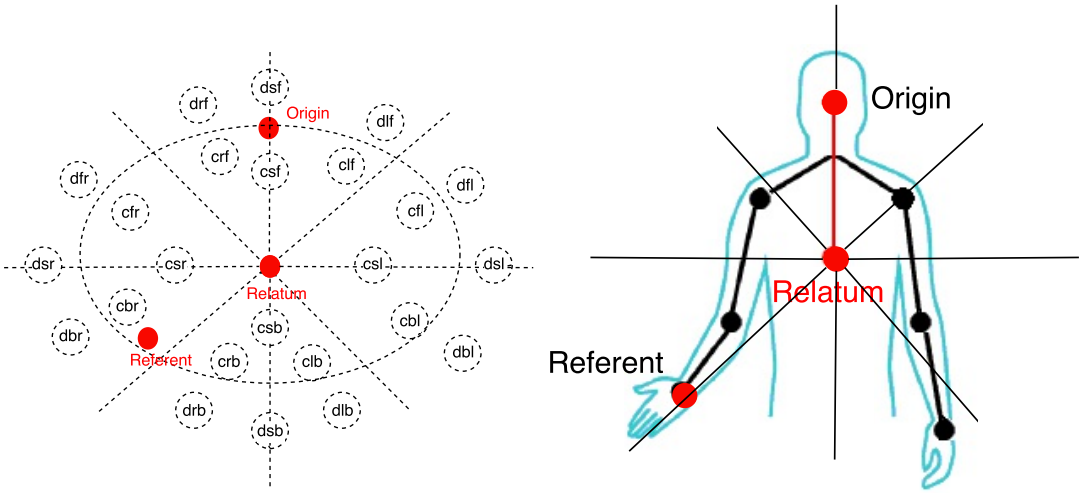


Figure 4.3: (left:) Full TPCC reference system. Relations are triplets of the letters $\{f, b, l, r, s, d, c\}$ which stand for: front, back, left, right, straight, distant, close, respectively. The system represents the position of the qualitative location of the *referent* relative to the *origin* and *relatum* (all shown as red points in 2D). In this example, the referent is considered close, back and right or "cbr". (right:) TPCC system applied to a human body pose. The *hand* body joint position is qualitatively described with respect to the origin and relatum which are fixed to the body joints *head* and *torso* respectively. The hand's location is represented as "cbr" with respect to head-torso line.

4.2.4 QSR Implementation

In this section, we give the implementation details of how the robot applies the three qualitative abstractions (QDC, QTC, TPCC) onto its human observations, resulting in a *QSR matrix*

$Q = [Q_{qdc}, Q_{qtc}, Q_{tpcc}]$, where each Q_i encodes a matrix of symbolic relations abstracted using the i^{th} calculus, where rows represent different sets of entities, and columns represent timepoints or poses detected.

Recall that the robot's quantitative observations are encoded as either a trajectory T, or human body pose sequence, S, as described throughout Chapter 3. That is, a trajectory is stored as a sequence of (x, y) detections, one per timepoint observed, and a human body pose as a vector of length 15 of (x, y, z) body joint locations, also one per timepoint. Further, the robot then requires a set of object locations, these can either be automatically segmented objects from its environment or manually defined static object locations. These objects are represented as single (x, y, z) coordinates in the metric map coordinate reference frame representing their centre point.

Trajectories

A trajectory is defined as $T = [t_1, t_2, \ldots, t_i, \ldots]$, where each t_i is the detected trajectory pose (x, y) at timepoint i. Figure 4.4 is an example of a trajectory as observed by the robot, and shown along with SOMa object locations in the environment (tables, chairs, plant pots). The process of computing a qualitative representation of a human trajectory is as follows:

- Given a new person detected by the robot, the person's location is translated into the map reference frame and a trajectory pose, t_1, is generated for the initial timepoint. A person model is added into the visualisation as per Figure 4.4 (left).
- The person is tracked through the environment. At each timepoint a trajectory pose is computed t_i. A purple line shows the evolution of the trajectory (until timepoint i) over a sequence of detections and can be seen in Figure 4.4 (centre).
- The trajectory observation ends as the person moves out of the sensor's field of view and cannot be detected any longer, shown in Figure 4.4 (right).

At each detected trajectory pose, a purple connected line forms a visualisation of the trajectory through the environment. This can be seen increasing in length as the number of observed poses increases as the person walks past the robot shown in Figure 4.4 (left to right).

For each trajectory pose, the pairwise qualitative calculi, i.e. QDC and QTC, symbolic relations are computed between the pose and surrounding objects (located in the same SOMa ROI as the pose).[1] This allows the robot to efficiently capture a spread of spatial relations throughout its environment resulting in a set (one per calculi) of pairwise relations between the trajectory pose and each object, at each timepoint. For example in Figure 4.4 (left), in the

[1] A trajectory pose comprises of a single 2D location (as opposed to multiple body locations), for this reason the robot does not apply a TPCC representation since there are insufficient points to represent the referent, origin and relatum.

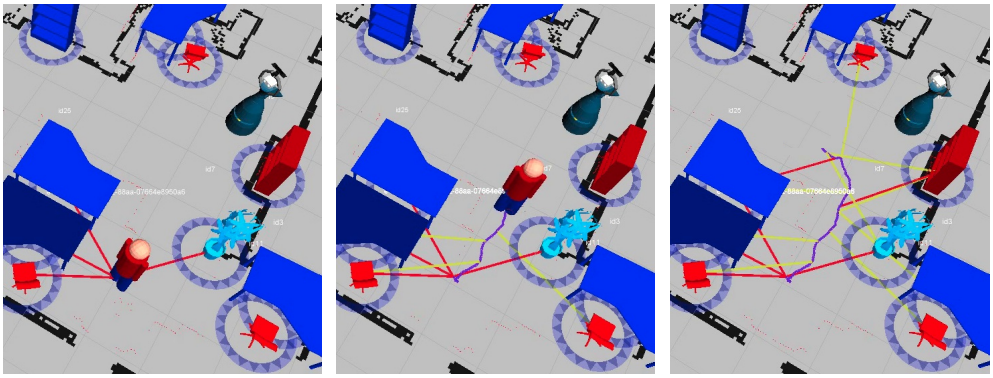


Figure 4.4: Implementing QSRs on a human trajectory relative to SOMa objects in the robot's environment. The purple line denotes the trajectory (x, y) poses in the map frame. Red and yellow lines represent the nearby selected objects used as entities to compute pairwise relations with the trajectory pose. Best viewed in colour.

absence of SOMa ROIs, the closest 4 objects to the initial trajectory pose are selected (and depicted by four connecting red lines between the initial trajectory pose and the four selected objects). Note, as a person moves through the environment, the closest objects to the trajectory pose are calculated for each new pose, and therefore can change as the person moves towards and away from different objects as in Figure 4.4. As the person walks past the robot, the objects closer to the robot (the chair, plant pot and bookcase to the right of the map) eventually get added to the set of objects and others are removed.

QDC: To compute the QDC sequence of relations, the distance between each trajectory pose and each of the objects is computed using Equation 4.1 and represented as a symbolic relation using the pre-defined threshold boundaries given by Δ.

QTC: For QTC relations, consecutive trajectory poses are taken and the relative motion is computed between the pose and the static objects, i.e. moving towards (-), away from (+) or stable (0).

This process results in two QSR matrices, $[Q_{qdc}, Q_{qtc}]$, where a row represents a vector of symbolic qualitative relations for a set of entities (trajectory-object pair), and each column represents a timepoint in the sequence of trajectory poses. For example, computing QDC relations for a trajectory of length m (number of poses), i.e. $T = [t_1, t_2, \ldots, t_m]$, results in a length $n \times m$ matrix of QDC relations Q_{qdc} (n rows refer to the different pairwise objects in the environment selected). For QTC, this results in a $(n \times m - 1)$ matrix of QTC relations, Q_{qtc}, since one QTC relation requires two consecutive trajectory poses no relation is computed for the initial trajectory pose t_1. When QDC and QTC relations are computed for the same trajectory observation, we often drop the QDC relation at t_1 to acquire two matrices of equal size Q_{qdc}

and Q_{qtc}. The two matrices can also be merged to create joint QDC and QTC relations for each pose-object pair, at each timepoint.

Human Pose Sequence

The process of computing a qualitative representation for a human body pose sequence is similar to that of a trajectory presented above. However, whereas a human trajectory consists of a sequence of poses, each pose being a single 2D location on the metric map plane, a human body sequence comprises of multiple human body poses which contain 15 (x, y, z) body joint locations relative to both camera frame and map frame. Still, the idea is very much similar; a set of symbolic QSR relations (QDC and QTC) are computed per key object, per timepoint, and a vector of TPCC relations is also computed, one per body pose/timepoint.

Consider a human body pose sequence of length m, i.e. $S = [p_1, p_2, \ldots, p_m]$, where each p_i is a human pose estimate, $p_i = [j_1, j_2, \ldots j_{15}]^T$, a vector of 15 (x, y, z) body joint locations. We describe how to abstract this encoded quantitative observation into multiple matrices of semantic qualitative symbols (one per calculi), starting with QDC and QTC since they both abstract from the map coordinate frame of reference, as opposed to TPCC which abstracts the 2D image plane locations.

QDC: To compute a matrix of QDC relations, Q_{qdc}, the distance between the location of each body joint pose and the key objects is computed using Equation 4.1. The distances are then represented with qualitative symbolic relations using the pre-defined threshold boundaries Δ, resulting in a $(15n \times m)$ matrix Q_{qdc}, where n is the set of all objects and m is the number of timepoints or poses.

QTC: Likewise, to compute QTC relations, the relative motion of a body joint location across consecutive human pose estimates is computed relative to each of the key static objects (in turn). This results in a $(15n \times m - 1)$ matrix Q_{qtc} of QTC relations (i.e. excluding the initial pose p_1). Similarly to the trajectory Q matrices, QDC and QTC relations can be merged. A simplified illustration of the two QSRs computed for a single human body pose can be seen in Figure 4.5. This shows the detected human body pose joints overlaid onto the person, and in the diagram we only consider QDC relations between the person's right hand and Object 1, and QTC relations between the left hand and Object 2.

TPCC: Qualitative TPCC relations are computed by abstracting the camera frame (i.e. 2D image plane) coordinates of the encoded human body pose sequence. Similar to literature, a 2-dimensional line is created from a pair of body joint locations, and the relative location of a third body joint is computed using the TPCC system shown in Figure 4.3 (left). Specifically, we fix the origin and relatum to specific body joints, namely the *head* and *torso* positions and compute the relative positions of each of the other body joints with respect to this line. An

illustration of the head-torso line overlaid onto the a single human body pose can be seen in Figure 4.3 (right). For a human body pose sequence of length m, a symbolic TPCC relation can be computed for each body joint relative to the head and torso (as they are fixed as the origin-relatum line), per timepoint. This results in a $(13 \times m)$ matrix Q_{tpcc} of TPCC relations (in practice fewer body joints locations are often used for computational efficiency).

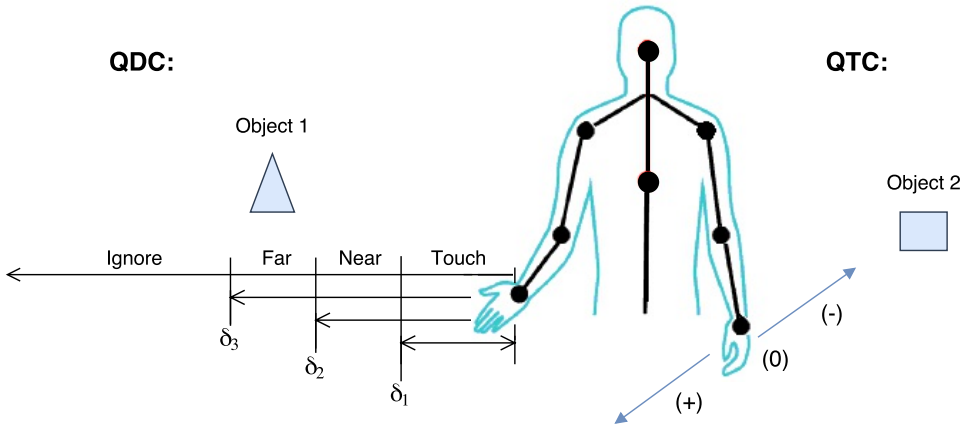


Figure 4.5: A QSR for a human body pose: A simplified illustration of QDC and QTC applied to a single human body pose. (left:) QDC (relative distance) between right hand and Object 1 (triangle), computed using $\Delta = [\delta_1, \delta_2, \delta_3]$. (right:) QTC (relative motion) between left hand and Object 2 (square). Blue arrows denotes possible motion towards $(-)$, or away from $(+)$ the static object.

Smoothing Filters

Two smoothing filters are proposed with the aim of reducing rapid switching of qualitative states between timepoints which can arise from visual noise, i.e. from quantitative sensor encodings. The two filter techniques are:

1. a 1-dimensional median filter,
2. a conceptual neighbourhood restriction applied to the relational state transitions.

The encoded QSR matrices obtained from the above abstractions are passed into a 1-dimensional median filter (one row at a time). Recall that one row in each matrix relates to a vector of length m (or $m-1$) relations which hold between a set of entities, where m is the number of timepoints encoded in the quantitative observation (poses). A 1-dimensional median filter has the effect of smoothing the vector of relational states obtained over a window of timepoints. This is performed for each row (i.e. each set of objects), for each calculi separately, e.g. each row in each Q matrix separately.

As an example, consider a human trajectory consisting of five poses ($m = 5$), $T = [t_1, t_2, t_3, t_4, t_5]$, and a single key object, o_1, in the environment. The distance at each pose,

$d_i = dist(t_i, o_1)$, is calculated as per Equation 4.1 and we consider three boundary thresholds, $\Delta = [\delta_1, \delta_2, \delta_3]$, which create four regions of space with natural language names: "touch", "near", "far" and "ignore". Given only a single object, only one trajectory-object pair is computed, i.e. $n = 1$, over the five poses, so the matrix Q_{qdc} has size (1×5). If the first (and only) vector of relational states contains rapid switching of states, e.g. the first row in Q_{qdc} is equal to the vector: $[touch, touch, far, touch, touch]$. A median filter passes a sliding window (with pre-defined size) over the vector and replaces one element at a time with the most common element currently within the window. Then the window moves one positional index and repeats. After the filter is applied (with a window > 1), the resulting vector will be smoothed, e.g. $[touch, touch, touch, touch, touch]$. This is a useful feature of the median filter. The robot uses a relatively small window size as to not remove important interactions occurring over a small number of timepoints, but still smooths the vector of QSR relations (the exact window size is given in the experimental Chapter 6).

The second smoothing technique is based upon observing valid relational transitions over consecutive timepoints. This is explained in detail for vectors of QTC relations in [Dondrup et al., 2014], and the technique was first introduced in [Fernyhough, 1997, Fernyhough et al., 1998]. The main idea is to obtain row vectors in Q_{qtc} consisting of sequences of QTC relations that satisfy valid transitions with respect to a conceptual neighbourhood graph (such as the one presented in Figure 4.1). The filter restricts state transitions that could not have occurred in continuous space. For example, an object can not be "moving away from" and then "moving towards" another object in consecutive frames, without first passing into a the "static" relative state. In practice, this relies on the sampling rate of the sensors in order to capture accurate quantitative observations that can be represented by the qualitative states. When the robot obtains a sequence that exhibit prohibited transitions, the reasonable interpretation is that a sensing discrepancy caused the quantitative measurement of the pose to be incorrect, or that the frame rate was too slow and missed the transition. This is particularly likely to be true for the QTC '0' state, e.g. consider a bouncing ball which will only instantaneously be represented as '0' relative to the ground. If a qualitative state is missing then it can be inserted. If multiple consecutive states are missing, the whole sequence can be ignored [Fernyhough et al., 2000].

4.2.5 Summary

To summarise this section, we have presented three viewpoint invariant QSR calculi that can efficiently represent the qualitative motion of observed humans as encoded as trajectories or body pose sequences from a mobile robot. Given recent literature and survey papers in the area of qualitative representations [Chen et al., 2015], the three QSR calculi introduced are each considered appropriate to encode different spatial aspects of the observations in a qualitative manner. However, it is not an exhaustive list and other qualitative calculi could be explored. Selecting the most appropriate qualitative representations is an open research problem; we hypothesise that this is task dependent, where a sufficiently rich qualitative representation is

required to encode discriminative information in order to draw similarities within the encoded data. This has been somewhat investigated in [Young and Hawes, 2015] but remains an open question.

Each of the QSR matrices Q, along with the filtering techniques are computed using the publicly available ROS library we co-authored QSRLib [Gatsoulis et al., 2016a,b]. In the experimental Chapter 6, we investigate the effect of using different combinations of the three presented QSRs to better understand which are most discriminative with respect to human motion patterns and simple activities.

4.3 Qualitative Time Series

One hypothesis in this thesis is that many human activities can be explained by a sequence of primitive actions over some duration of time. In order to learn these sequences, as patterns within the spatially abstracted data, the robot must temporally abstract the observations also. The QSR matrices, introduced in the previous section, represent multiple observed qualitative relations holding between entities for each pose or timepoint. We consider these as a time series of observational data.

In this section, we extract compact sequences of relations from these vectors. This is achieved by compressing repeated QSR relations which hold between entities, i.e. when the relation is stable for some period of time. The resulting encoding is the *interval representation*, which can be described as a temporally connected set of semantic intervals, each maintaining a duration of time over which they hold. This representation is an abstraction of a Qualitative Spatial-Temporal Activity Graph (QSTAG) first introduced in [Duckworth et al., 2016a], and is closely related to an intermediate representation we developed in [Duckworth et al., 2016b, Gatsoulis et al., 2016a,b].

4.3.1 Interval Representation

Consider a human trajectory encoded as a Q_{qtc} matrix, relative to a single object in the environment, i.e. $n = 1$. If the trajectory appears to be moving towards Object 1, o_1, (QTC relation: '$-$'), for some consecutive number of frames τ, and then is static (0) with respect to that object for τ' further frames, we can compress the first row in Q_{qtc} into an interval representation consisting of two intervals: $I = \{\iota_1, \iota_2\}$ where each interval can be represented as a tuple, $\iota_1 = (\text{'}-\text{'}, [0, \tau - 1])$, and $\iota_2 = (\text{'0'}, [\tau, \tau + \tau' - 1])$. Each interval $\iota \in I$ maintains a QSR value and the start and end time points of the interval duration which the relation held. We introduce the terminology *lesser end point* as the start time point of an interval, ι, and represent it by ι^-, and similarly, the *greater end point* as the end point of the interval and represent it by ι^+, where $\iota^- < \iota^+$.

The simple two interval examples above can be interpreted visually and is shown in Figure 4.6. It represents the qualitative spatial relations between the trajectory poses and a single

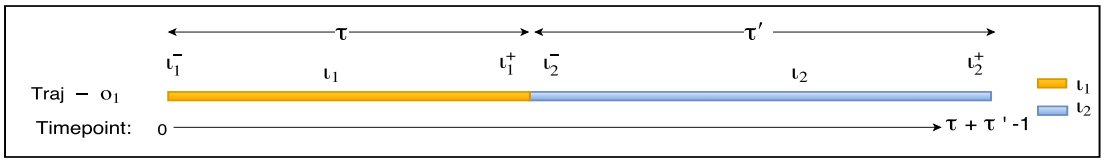


Figure 4.6: Interval representation of two intervals $I = \{\iota_1, \iota_2\}$. The intervals maintain the QSR relation that holds between a trajectory and a single object, o_1. Best viewed in colour.

object o_1. Here, the x-axis represents discrete time (starting at 0 until $\tau + \tau' - 1$), the first interval ι_1 is represented as a yellow duration, $(\iota_1^-, \iota_1^+) = (0, \tau - 1)$, and the second interval ι_2 is blue with $(\iota_2^-, \iota_2^+) = (\tau, \tau + \tau' - 1)$.

Usually, a QSR matrix contains more than a single row, i.e. it contains n rows where each represents the trajectory relative to a key object in the environment, or up to $15n$ rows if each human body joint is encoded with each of the n objects (in the case of a human body pose sequence with 15 joints). Two rows encoded from an encoded Q matrix representation of a human body pose sequence might look something like Figure 4.7. It depicts relations between two human body joints, the left hand and the right hand, each with respect to a single object o_1, and shows a set of five intervals. Using an interval representation facilitates the use of temporal abstraction techniques which allow the robot to extract discrete descriptors encoding temporal relations of pairwise intervals. These discrete descriptors are introduced in the next section and are used to compare multiple observations.

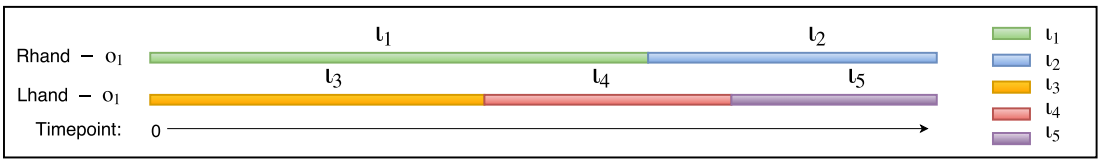


Figure 4.7: Interval representation of five intervals $I = \{\iota_1, \iota_2, \iota_3, \iota_4, \iota_5\}$. Pairwise QSR relations between right hand body joint and Object 1 (Rhand $- o_1$), and left hand body joint and Object 1 (Lhand $- o_1$). x-axis represents discrete timepoints. Best viewed in colour.

Interval Terminology

Here we define some terminology relating to an interval representation that will become useful in the next section. Given any two intervals in an interval representation, $\iota_1, \iota_2 \in I$, we say they are *temporally connected* if there exists no temporal break between the start of the first interval and the end of the second interval (defined for discrete time). This can be checked using the following inequalities:

$$\text{(a)} \;\; \iota_1^+ < \iota_2^- - 1,$$

(b) $\iota_1^- - 1 > \iota_2^+$.

If either (a) or (b) holds true, then there exists a temporal gap between ι_1 and ι_2 (of at least one discrete timepoint). This is equivalent to checking whether part of either interval occurred at the same time or consecutively (with no temporal gap). For example, in Figure 4.7, intervals ι_1 and ι_3 are temporally connected since at least part of the interval overlap; whereas ι_3 and ι_5 are not temporally connected since there is a period of time, namely during ι_4, which occurs between them. Given any set of intervals defined as an interval representation, I, we can also define the earliest starting timepoint as ι^{--}, and the final end timepoint as ι^{++}. That is, $\iota^{--} = min(\{\iota^- \mid \forall\ \iota \in I\})$ and $\iota^{++} = max(\{\iota^+ \mid \forall\ \iota \in I\})$. This can be thought of as the limits of the x-axis in the interval representation visualised in Figure 4.6. For example, $\iota^{--} = 0$ and $\iota^{++} = (\tau + \tau' - 1)$.

Finally, we define two sets of intervals known as the *starting intervals set*, I^-, and *ending intervals set*, I^+. An interval is a member of either set if it shares an interval endpoint with the observation's start or end timepoints ι^{--} or ι^{++} respectively. That is:

$$I^- = \{j \mid \forall j \in I, j^- = \iota^{--}\},$$
$$I^+ = \{j \mid \forall j \in I, j^- = \iota^{--}\}$$

For example, in Figure 4.7, intervals ι_1 and $\iota_3 \in I$ both occur at the start of the interval representation, i.e. $\iota_1^- = \iota_3^- = \iota^{--}$. Therefore both are in the starting intervals set, $I^- = \{\iota_1, \iota_3\}$. Similarly, the ending intervals set $I^+ = \{\iota_2, \iota_5\}$ since $\iota_2^+ = \iota_5^+ = \iota^{++}$. For intervals in these two sets, we do not observe the actual start or end time of the QSR sequence encoded. This is particularly important when we temporally abstract the intervals in the next sections.

4.3.2 Qualitative Temporal Representation

In the previous section we introduced multiple qualitative *spatial* representations the robot uses to abstract spatial aspects of human observations (trajectories or human body pose sequences). Here, we introduce a *qualitative temporal representation* the robot uses to abstract away the exact temporal details of an observation. For example a human activity such as "talking on the phone" can greatly vary in duration, however, abstracting away the exact duration, the components of the activity, such as "hand approaches phone", "hand raises phone", etc. may have a relatively similar order. That is, the sequence of encoded qualitative spatial relations maintains an ordering sufficient to describe the activity taking place.

The techniques introduced in this section abstract the exact temporal aspects of observed QSR sequences as encoded in an interval representation. This allows the robot to compare across multiple observations and obtain partially ordered sequences of relations which hold between observed entities. To perform the temporal abstraction, the robot uses Allen's Interval Algebra (IA) [Allen, 1983] which defines 13 possible qualitative temporal base relations between any pair of intervals ι_1 and ι_2. Possible temporal relations include: ι_1 *before* ι_2, ι_1 *after* ι_2, ι_1

meets ι_2, ι_1 *overlaps* ι_2, etc.. All 13 possible IA relations are shown in Table 4.8a, along with the inverse relations (which remains the same for "equals"). The table also illustrates the 7 temporal interpretations of the possible IA relations between the two intervals ι_1 and ι_2.

An alternative way to represent the Allen Interval Algebra relation between two intervals ι_1 and ι_2, is to use the endpoints values (ι_1^-, ι_1^+) and (ι_2^-, ι_2^+) to define the temporal relations, such as shown in Table 4.8b.

4.3.3 Interval Graph

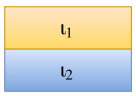
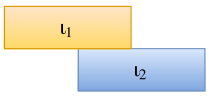
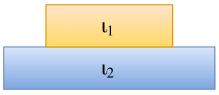
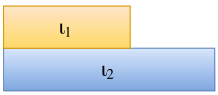
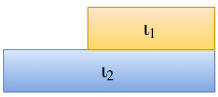
Graph Language

A *simple graph* $G = (V(G), E(G))$, with p vertices and q edges consists of a *vertex set* (or *node set*) $V(G) = \{v_1, v_2, \ldots, v_p\}$ and an *edge set* (or *arc set*) $E(G) = \{e_1, e_2, \ldots, e_q\}$ where each edge is an unordered pair of vertices. An edge $e = \{u, v\}$ can be denoted uv or vu. *Directed graphs* are simple graphs where edges are ordered pairs of vertices, such as, an edge $e = [u, v]$. The *degree* of a vertex v in graph G is denoted $d_G(v)$ and is the number of edges that are incident to v.

Interval Graph

Given a set of intervals defined as an interval representation, I, we can apply Allen's Interval Algebra in order to abstract away the exact temporal details of the observation. Between any pair of intervals, $\iota_1, \iota_2 \in I$, we can compute the IA relation that holds and abstract away the quantitative duration of the two intervals. For example, in Figure 4.7, the IA relation that holds between intervals ι_1 and ι_2 is *meets*. This allows us to semantically link ι_1 and ι_2 with a temporal relation *meets* and ignore the exact quantitative interval endpoints, (ι_1^-, ι_1^+) and (ι_2^-, ι_2^+).

By encoding a graph node for each interval and an IA relation for every pair of intervals, we are able to create a semantic *interval graph* representation g [de Ridder et al., 2016]. A node in the interval graph, $\iota' \in g$, represents an interval $\iota \in I$ and abstracts away the temporal information. A node encodes the entities (objects) and the QSR value that holds during the corresponding interval, i.e. the exact timepoints are ignored in the node. Directed arcs are labelled in order to represent the IA relation that holds between any temporally connected intervals; recall this excludes pairwise intervals where a temporal gap exists between them. This restriction has the effect of removing arcs with IA relations *before* or *after* from the interval graph. The reason for not encoding these arcs is that they represent redundant information, e.g. given three intervals with positive durations, $\iota_1, \iota_2, \iota_3 \in I$, if ι_1 *meets* ι_2 and ι_2 *meets* ι_3, temporal logic states that ι_1 is *before* ι_3, and maintaining all three arcs in this situation is unnecessary. The output is an interval graph with fewer arcs, but that maintains the same qualitative information. Lastly, we do not compute an IA relation between pairwise intervals if they both belong to either the starting intervals set I^-, or both to the ending intervals

Relation	Inverse	Illustration	Interpretation
ι_1 before ι_2	ι_2 after ι_1		ι_1 takes place before ι_2.
ι_1 equal ι_2	ι_2 equal ι_1		ι_1 is equal to ι_2.
ι_1 meets ι_2	ι_2 met by ι_1		ι_1 meets ι_2.
ι_1 overlaps ι_2	ι_2 overlapped by ι_1		ι_1 overlaps with ι_2.
ι_1 during ι_2	ι_2 during inverse ι_1		ι_1 takes place during ι_2.
ι_1 starts ι_2	ι_2 started by ι_1		ι_1 starts at the same time as ι_2, but finishes first.
ι_1 finished ι_2	ι_2 finished by ι_1		ι_1 finishes at the same time as ι_2, but starts later.

(a) Allen's Interval Algebra based upon two intervals ι_1 and ι_2. Showing all 13 possible relations, 6 relations have an inverse, along with qualitative interpretations.

IA Relation	Equivalent endpoints definition
ι_1 before ι_2	$\iota_1^+ < \iota_2^- - 1$
ι_1 equals ι_2	$(\iota_1^- = \iota_2^-)$ & $(\iota_1^+ = \iota_2^+)$
ι_1 meets ι_2	$\iota_1^+ = \iota_2^- - 1$
ι_1 overlaps ι_2	$(\iota_1^- < \iota_2^-)$ & $(\iota_1^+ > \iota_2^-)$ & $(\iota_1^+ < \iota_2^+)$
ι_1 during ι_2	$((\iota_1^- > \iota_2^-)$ & $(\iota_1^+ \leq \iota_2^+))$ or $((\iota_1^- \geq \iota_2^-)$ & $(\iota_1^+ < t_2^+))$
ι_1 starts ι_2	$(\iota_1^- = \iota_2^-)$ & $(\iota_1^+ \neq \iota_2^+)$
ι_1 finishes ι_2	$(\iota_1^+ = \iota_2^+)$ & $(\iota_1^- \neq \iota_2^-)$

(b) Allen's Interval Algebra defined on endpoint intervals, (ι_1^-, ι_1^+) and (ι_2^-, ι_2^+), assuming discrete time.

Figure 4.8: Allen's Interval Algebra [Allen, 1983] as represented using discrete time.

set I^+. This is a somewhat similar assumption to when encoding an Activity Graph in the literature [Sridhar, 2010, Duckworth et al., 2016b]. In these cases, there is insufficient temporal information to compute the pairwise temporal relation since the interval's start or end points are not encoded within the observation.

An example interval graph can be seen in Figure 4.9, which encodes five nodes, one per interval present in Figure 4.7. Examining one interval, ι_1, in more detail: the interval is represented by the node ι'_1 and contains the object information (*Rhand*, O_1) and the spatial relation that holds during ι_1, i.e. the information is temporally abstracted from the corresponding interval. The set of temporally connected intervals to ι_1 is the set $\{\iota_2, \iota_3, \iota_4\}$, and the IA relations that hold between ι_1 and ι_2 is *meets*; and between ι_1 and ι_4 is *overlaps*; these labels can be seen on the directed arcs between the corresponding nodes. As introduced earlier, $I^- = \{\iota_1, \iota_3\}$ and $I^+ = \{\iota_2, \iota_5\}$, so there are no arcs encoded between the corresponding nodes, i.e. no arc between ι'_1 and ι'_3 even though the intervals are temporally connected.

Note, the direction of the arc is significant to represent the order of the arguments, e.g. reversing the arc between ι'_1 and ι'_2, would reverse the order of the arguments and therefore the IA relation that holds would be the inverse temporal relation *met by*, since ι'_2 is *met by* ι'_1.

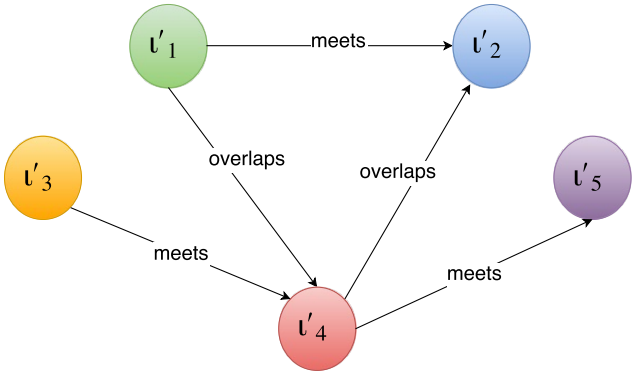


Figure 4.9: Interval graph, g, encoded from interval representation I in Figure 4.7. A node ι'_i represents a temporally abstracted interval $\iota_i \in I$. Connecting directed arcs represent the IA relation that holds between pairwise intervals which are temporally connected. No arc is computed between two intervals if both are in the starting or both in the ending interval sets, I^- or I^+. Best viewed in colour.

An interval graph can be computed from an interval representation by performing the following steps:

1. A node, $\iota' \in g$, is created to represent each interval $\iota \in I$. It maintains the object and QSR relation information from the corresponding interval, resulting in $|I|$ nodes.

2. Using the endpoints of all intervals, compute the values $\iota^{--} = min(\{\iota^- \mid \forall \iota \in I\})$ and $\iota^{++} = max(\{\iota^+ \mid \forall \iota \in I\})$.

3. For every pair of intervals $\iota_i, \iota_j \in I$, the following operations are performed:

   (a) Check if $(\iota_i^+ < \iota_j^- - 1)$ or $(\iota_i^- - 1 > \iota_j^+)$, i.e. whether the two intervals are temporally connected or not.

   (b) Check if $\iota_i^- = \iota_j^- = \iota^{--}$ or $\iota_i^+ = \iota_j^+ = \iota^{++}$, i.e. if both intervals are members of the starting or ending intervals sets.

   (c) If (a) and (b) are both false, the IA relation that holds between ι_i and ι_j is calculated using Table 4.8a.

   (d) Check if the IA relation is an inverse relation, i.e. one of those in the second column of Figure 4.8a. If so, reverse the ordering of ι_i and ι_j and invert the IA relation.

   (e) Finally, add the arc between the two nodes ι_i' and ι_j' to the interval graph with the corresponding label.

Performing the above steps produces a compact qualitative interval graph representation of a human observation. In the next section, we describe how an interval graph can be discretised into a collection of novel qualitative descriptors where each descriptor captures short, partially overlapping sequences of relations and is used to represent the observation.

4.4 Qualitative Descriptors

Given an observation of a human encoded as an interval graph, the aim is to extract a set of sub-structures from the graph to efficiently represent it for the purpose of approximate graph comparison. The general idea is for each temporally-extended observation to be represented as a collection of sub-graphs which capture overlapping and semantically meaningful qualitative descriptors. The interval graph is ultimately represented as an efficient, low dimensional vector of numerical counts over this set of descriptors.

To do this, we first extract *graph paths* through the directed interval graph with constraints placed on the length and complexity of valid paths; these are defined as *qualitative descriptors*. The terms qualitative descriptor and graph path are used interchangeably in this section. Each graph path represents an actual observed sequence of temporally connected qualitative spatial relations holding over some interval of time within the observation. The extracted graph paths temporally overlap, and therefore maintain partial temporal ordering within the observation. The observation is then represented as an unordered collection (a vector of counts) of these overlapping descriptors known as a *bag-of-qualitative-words*. As an analogy, consider a text document that is encoded as a collection of natural language words and n-grams.

4.4.1 Extracting Graph Paths

Given an interval graph $g = (V(g), E(g))$ with p vertices (nodes) and q edges (arcs), the *vertex set* $V(g) = \{\iota_1', \iota_2', \ldots \iota_p'\}$ and the *edge set* $E(g) = \{e_1, e_2, \ldots, e_q\}$, where each edge is an ordered

pair of vertices, $e_i = [\iota'_k, \iota'_l]$ where $k \neq l$. Then, a *path* of length n is defined as a set of n nodes that form a path through g by following directed edges in $E(g)$. The nodes and edges in a path form a sub-graph of the original interval graph and is defined as a *graph path*, i.e. a graph path w is defined as:

$$w = \{\iota'_1, \iota'_2, \ldots, \iota'_n\} \text{ s.t. } \{\iota'_i, \iota'_{i+1}\} \in E(g), \forall \;\; 1 \leq i < n.$$

All graph paths up to some length η (≥ 0) are evaluated starting from each node in the interval graph in turn. This results in a collection (or bag) of potentially overlapping graph paths used to represent the encoded interval graph, e.g. $d = \{w^1, w^2, \ldots\}$.

In practice, the graph paths are often also limited to a maximum number, ρ, of encoded entities or objects pairs. Recall each node ι' in an interval graph g is a representation of the set of entities (objects) and the qualitative spatial relations that holds between them over the corresponding interval of time. Therefore we limit the graph paths extracted to contain at most ρ different sets of objects. If a large number of objects are encoded in an interval graph, then this reduces the number of possible graph paths extracted by restricting a path to include temporally connected intervals of relations between a subset of at most ρ object pairs. However, the combination of object pairs overlap in the interval representation and so implicitly the graph paths maintain information between all objects. As an example, the interval graph in Figure 4.9 can be represented as a collection of 13 graph paths; first by selecting the maximum path length and the maximum number of pairwise objects, e.g. $\eta = 2$ and $\rho = 2$. This produces the following set of graph paths:

- length 0-paths: $\{\iota'_1,\ \iota'_2,\ \iota'_3,\ \iota'_4,\ \iota'_5,\}$
- length 1-paths: $\{\ (\iota'_1\ \textit{meets}\ \iota'_2),\ (\iota'_1\ \textit{overlaps}\ \iota'_4),\ (\iota'_4\ \textit{overlaps}\ \iota'_2),\ (\iota'_3\ \textit{meets}\ \iota'_4),\ (\iota'_4\ \textit{meets}\ \iota'_5)\}$,
- length 2-paths: $\{(\iota'_1\ \textit{overlaps}\ \iota'_4\ \textit{meets}\ \iota'_5),\ (\iota'_3\ \textit{meets}\ \iota'_4\ \textit{overlaps}\ \iota'_2),\ (\iota'_3\ \textit{meets}\ \iota'_4\ \textit{meets}\ \iota'_5)\}$.

The extracted graph paths can be more easily visualised as in Figure 4.10. Here: (a) represents the graph paths of length 0, i.e. no arcs and one node; (b) represents paths of one arc and two nodes; and (c) represents paths extracted with two arcs and three nodes. Note, there are only two object pairs encoded in this example interval graph (and interval representation), so there is no paths restricted by setting ρ.

The total number of graph paths extracted from an interval graph depends upon multiple factors: the number of objects encoded within the interval representation, i.e. the number of rows of the QSR matrix Q; the QSR calculi used along with the values they can take; the path-length limit η; and the maximum number of pairwise objects limit ρ. These selections are given in the later experimental sections.

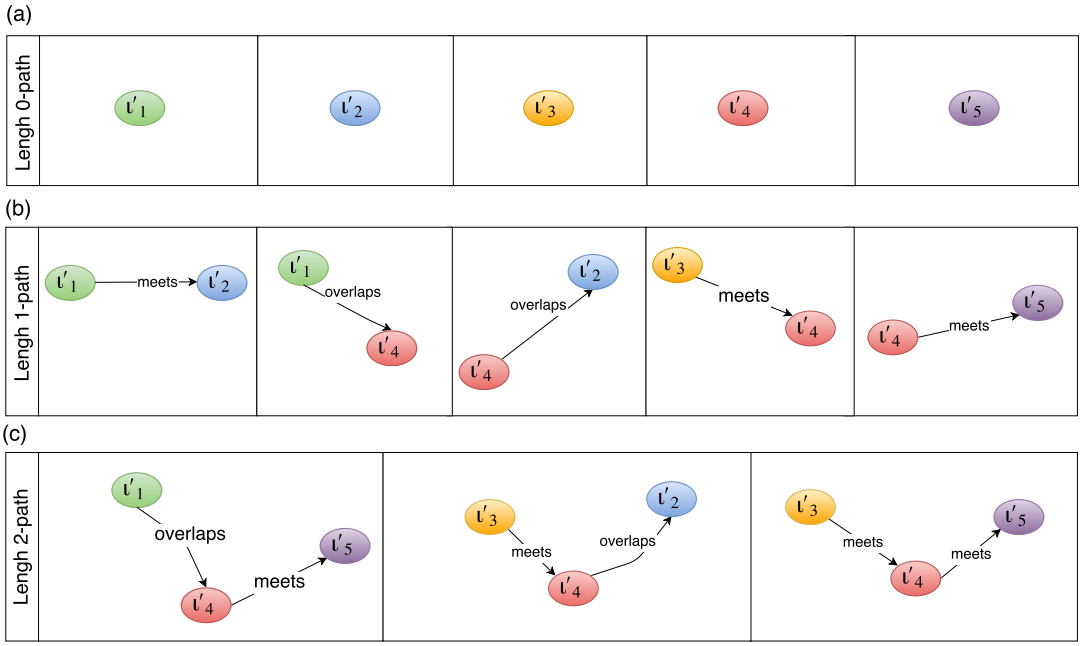


Figure 4.10: All graph paths extracted from the interval graph shown in Figure 4.9. (a) Paths of length 0. (b) Paths of length 1 (one arc, two nodes). (c) Paths of length 2 (two arcs, three nodes).

Once all graph paths have been extracted from an interval graph, the observation can be summarised by a bag of its extracted paths, e.g. $d = \{w^1, w^2, \ldots, w^{13}\}$, where the length depends upon the number of graph paths extracted; akin to the collection of words that make up a document. The collection of graph paths can be thought of as a *bag-of-qualitative-words* representation of the abstracted observation, where each word represents a graph path structure. In the traditional bag-of-words, word ordering is lost, as an unordered collection of each word is taken and added to the so called bag. However, in our technique, each descriptor contains more semantic information than a single word, that is, each of our descriptors is a graph path that represents a sequence of temporally connected QSR intervals. There are two important distinctions:

1. our graph paths encode multiple timepoints of qualitative relations. That is, they maintain local ordinal information about qualitative spatial relations occurring within a temporal slice of an observation.

2. setting the maximum path length greater than one, the graph paths extracted from an interval graph overlap. That is, individual nodes in a graph can be present in multiple graph paths, inherently maintaining structure.

Therefore, an unordered bag of extracted graph paths inherently encodes more long-term tem-

poral structure. This is a key feature of our bag-of-qualitative-words representation and is crucial to learning patterns across multiple observations.

4.5 Concluding Remarks

To summarise this chapter, each quantitative human observation is first abstracted as a sequence of qualitative *spatial* relations and further abstracted temporally using Interval Algebra. This is performed to reduce the dimensionality of the observed data and allow for greater generalisation between multiple observations. Each observation is encoded as a novel interval graph representation, where all graph paths are extracted and a set of discrete descriptors are extracted. These partially overlapping qualitative descriptors are used to represent the observation as a bag-of-qualitative-words, where constituent words are overlapping graph paths maintaining local temporal structure.

In the next chapter, we accumulate multiple human observations using these techniques and learn a code book which can be considered as a discrete vocabulary used to discretise all observations. We then present methods for learning common patterns within the encoded observations, which are considered as human activity classes with which to describe the observations. This process results in a low dimensional representation of the collection of human observations, each as a distribution over the learned activities.

Chapter 5

Extracting Activity Classes with Unsupervised Techniques

In this chapter, we build upon the representations of observed human activities as encoded using techniques introduced in the previous chapters. We show that by accumulating multiple observations into this representation, the robot is able to learn common and repeating patterns in the data in an unsupervised setting. This is achieved by tasking the mobile robot to patrol and observe a human populated environment for long durations of time, which allows the robot to:

1. recover semantically meaningful classes from its observations; where each class is considered as a human motion or activity class.

2. represent each human observation as an efficient, low dimensional vector over the set of learned classes that take place in the environment. Depending upon technique, this is represented as either a simple class assignment, or as a probabilistic mixture over classes.

As introduced in the previous chapter, the robot is able to discretise a human observation into a collection of partially overlapping, qualitative descriptors by first encoding an interval graph representation and then extracting discrete descriptors as graph paths. This process is extended to discretise a collection of human observations where all observed unique graph paths form a vocabulary; all unique graph paths extracted from a collection of interval graphs are combined into one set to create a discrete vocabulary defined as a *code book*. Each human observation is then encoded as a histogram (numerical counts of occurrences) of each constituent *code word*, and a vector representation is computed as a sparse *activity histogram* over the code book.

Given this discrete representation, the robot makes use of multiple unsupervised learning techniques in order to recover a set of latent activity classes from the human observations; these include: a standard unsupervised clustering technique; linear algebra low-rank approximation;

and a generative probabilistic method. Specifically, the information retrieval techniques take inspiration from the field of text analysis, where they were developed for learning patterns in large corpora of text documents. The analogy is that one human observation encoded by the robot is considered akin to a document comprised of basic discrete units, such as words. The robot's aim is then to recover a set of latent classes, concepts or topics, to best represent the collection of observations. The terms classes, concepts or topics can be used interchangeably throughout.

We first describe the process of converting a collection of interval graphs into a discrete representation, and provide an extended example for clarity. Then briefly introduce the unsupervised techniques employed and motivate their selection. Experiments and results are presented on three real world datasets in the experiments chapter next.

5.1 Discrete Histogram Representation

5.1.1 Code Book

The robot represents an observation of a human as a collection of its constituent discrete descriptors (graph paths) as described in Secion 4.4.1. That is, given an observation encoded as an interval graph, the robot extracts a bag of overlapping graph paths, where paths are used to represent the interval graph for the purpose of graph comparison. Formally, given an interval graph $g = (V(g), E(g))$ comprising of vertices V and edges E, each graph path is defined as a set of vertices $w = \{\iota'_1, \iota'_2, \dots, \iota'_n\}$ s.t. $[\iota'_i, \iota'_{i+1}] \in E(g), \forall \;\; 1 \leq i < n$, and is represented as a bag of discrete descriptors $d = \{w^1, w^2, \dots\}$. The set of unique graph paths V_d is then defined as:

$$V_d = \bigcup_{w^i \in d} w^i.$$

We extend this to a collection of M observations, where each observation is encoded as an interval graph g, i.e. $G = [g_1, g_2 \dots, g_M]$. For each interval graph, a bag of graph paths d is computed known as a *bag-of-qualitative-words*, i.e. $D = \{d_1, d_2 \dots, d_M\}$, and a discrete vocabulary known as a *code book*, $\mathcal{V}_D$ is automatically generated. Formally, we write:

$$\mathcal{V}_D = \bigcup_{d \in D} V_d,$$

where each graph path $w_i \in \mathcal{V}_D$ is unique and defined to be a *code word*, i.e. $\mathcal{V}_D = \{w_1, w_2, \dots, w_N\}$ is the *code book*. Note that we use subscripts to denote the index of a code word, w_i, in the code book and superscripts, w^j, to indicate an observed graph path in a bag-of-qualitative-words.

The code book represents the unique set of graph paths extracted from all the observed

interval graphs where the total number of them, N, is not known or fixed in advance.[1] Note, the code book specifically depends upon the qualitative descriptors extracted from the encoded interval graphs, i.e. altering the encoded observations, or changing the graph path parameters may result in a different code book and different activity histograms.

In order to compare two graph paths together, a distance based graph kernel is used to approximately represent each as a numeric value, and is described in detail in [Frasconi et al., 2014]. This reduces the graph matching and graph isomorphism checking to an efficient integer comparisons which checks whether two graph paths have identical hash values. This graph hashing technique is implemented for each graph path in the open source software library, QSRLib [Gatsoulis et al., 2016a,b].

As an alternative method, a code book could be computed in advance. One can compute all possible combinations of qualitative relations and intervals, between all possible object combinations. However, this is a less efficient process and the resulting code book is much larger with a number of redundant code words, i.e. code words that are never observed. This ultimately results in a more sparse feature space. This is not considered suitable for the task.

5.1.2 Activity Histogram

Given a code book of length N encoded from a collection of M human observations, each observation is represented as an N-dimensional feature vector describing the frequency of each code word in its bag-of-qualitative-words creating an *activity histogram*, h, over the code book $\mathcal{V}_D$. In this case, the activity histogram is considered a sparse feature vector representation since it may contain many zero code word counts. The representation is similar to a bag-of-words, where code words ignore positional arrangement; however the code words themselves are partially overlapping sequences of temporally connected QSR intervals and therefore maintain local temporal structure within the observation.

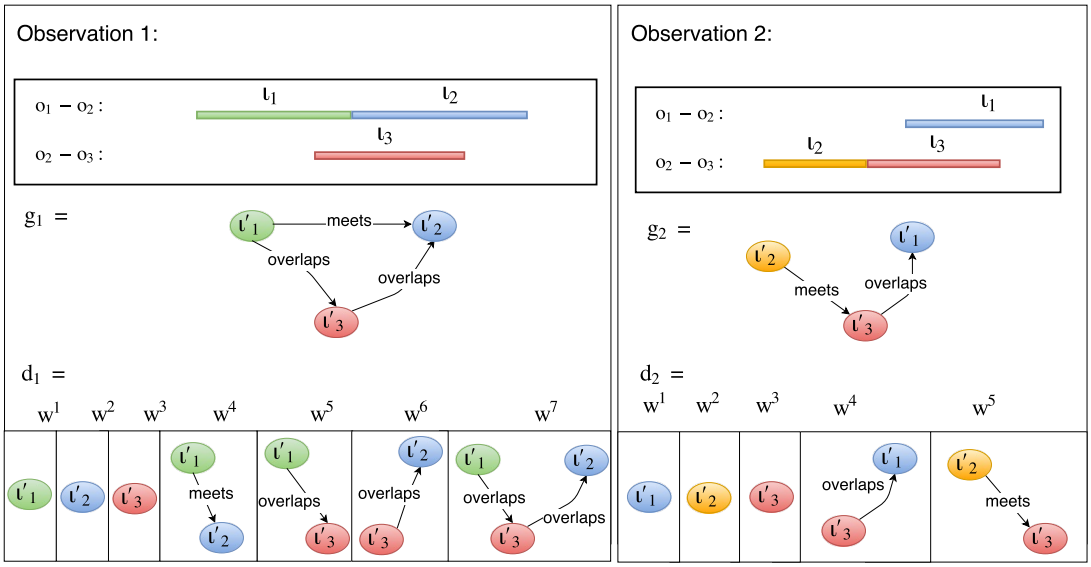
The activity histograms are each represented as N-dimensional vectors of counts, and therefore, similarly to a bag-of-words method, they are vertically stacked to produce a *term-frequency matrix* C which represents the entire corpus of observations as an efficient $M \times N$ matrix, where M is the number of rows corresponding to the number of observations encoded; and N is the number of unique code words corresponding to the length of the code book, e.g. $C = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_M]$, where each $\mathbf{h}_i$ is an N-dimensional vector of counts.

5.1.3 Extended Example

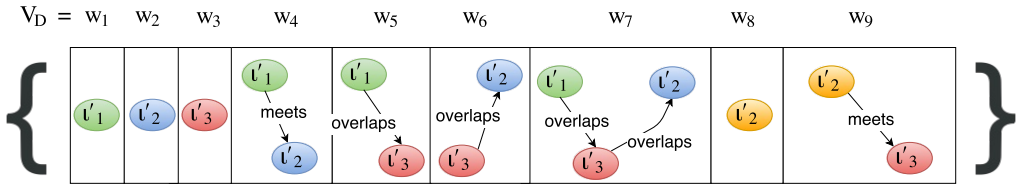
As an example of the above process, suppose the robot makes two observations and abstracts each instance by encoding the qualitative spatial representation between two common pairs of entities, e.g. (o_1, o_2) and (o_2, o_3), where each could relate to a body joint or object in

[1] In Chapter 7 we introduce techniques to dynamically update the code book given incremental observations, i.e. N increases with newly observed code words.

the environment. Further suppose that each observation is encoded as a sequence of QSRs represented by three intervals, $\{\iota_1, \iota_2, \iota_3\}$, between the two pairs of objects. For example, given a spatial calculus comprising of four relations, denoted by *green, blue, red* and *yellow* in Figure 5.1a, the first observation could be encoded as: (o_1, o_2) exhibit a *green* relation, followed by the *blue* relation, whilst (o_2, o_3) exhibit a *red* spatial relation. This example, along with a second observation, are encoded as interval representations and can be seen in Figure 5.1a (top). The specific spatial relation is depicted by the colour of the intervals, however any pairwise qualitative spatial calculi could be used here.



(a) (top:) Two example observations as encoded as QSR interval representations. (middle:) Corresponding two interval graph representations g_1 and g_2. (lower:) All possible graph paths extracted from the two interval graphs, each as a bag-of-qualitative-words d_1 and d_2.



(b) Code book, $\mathcal{V}_D$, generated by taking the union of the two bags of extracted code words $\{d_1, d_2\}$.

Figure 5.1: Extended representation example. The specific spatial relational value is depicted by the colour of the interval/nodes. Best viewed in colour.

The robot encodes the two corresponding interval graphs, $G = [g_1, g_2]$, by computing IA relations between pairwise intervals (shown middle); and (lower) all discrete qualitative descriptors are extracted as graph paths up to length 2, i.e. η and ρ are both ≥ 2 . It can be seen

that g_1 is represented as seven graph paths, and g_2 as five, i.e. $d_1 = \{w^1, w^2, \ldots, w^7\}$, and $d_2 = \{w^1, w^2, \ldots, w^5\}$, where $D = [d_1, d_2]$ represents the two bags-of-qualitative-words.

We remind the reader that the information maintained within an interval graph node ι' includes: 1) the QSR relation that holds over the corresponding interval ι, and 2) the set of objects the relations hold between. Inspecting the two bags-of-qualitative-words, d_1 and d_2, we see that they share particular graph paths (where the object pair and QSR relation matches between all nodes and edges), namely: $w^6 \in d_1$ is the same descriptor as $w^4 \in d_2$. This descriptor relates to the spatial relation defined by *blue* holding for an interval of time between entities o_1 and o_2, and "overlapped by" the *red* spatial relation exhibited by entities o_2 and o_3. This is a common qualitative sequence between the two observations, and therefore a common descriptor.

Taking the union of the two bags forms a code book $\mathcal{V}_D$ of unique code words. This is shown in Figure 5.1b, where $\mathcal{V}_D = [w_1, \ldots, w_9]$ comprising of nine unique code words. Each bag-of-qualitative-words is encoded as an activity histogram by counting the occurrence of each code word in each bag. Finally, the activity histograms are vertically stacked to generate a term-frequency matrix C, e.g. in our example C is a (2×9) matrix:

$$C = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}.$$

Encoding multiple observations into such a term-frequency matrix allows the robot to compare them and recover latent structure. In the next sections, we describe increasingly complex learning methodologies which are applied to the encoded term-frequency matrix in order to learn patterns in the data, and ultimately, classes of human behaviours and activities.

5.2 Activities by Unsupervised Clustering

A simple unsupervised clustering technique can be used to learn patterns embedded in an encoded term-frequency matrix. The aim is to cluster together repeated structure encoded as qualitative descriptors in similar observations. The clustering operation allows the robot to represent many observations by their emergent class, where each observation is assigned to a single activity class, and each activity class can be summarised by a single point in the N-dimensional training feature space. In the next sections we introduce methods where these assumptions are less restrictive.

5.2.1 k-means Algorithm

One such simple unsupervised clustering technique is the k-means algorithm [MacQueen et al., 1967]. It categorises data into disjoint clusters by trying to separate samples into groups of equal variance. The algorithm is parametrised by the number of clusters and requires this value,

k, to be specified in advance.

Given an $(M \times N)$ term-frequency matrix C, the k-means algorithm analyses the collection of M samples and learns k disjoint N-dimensional clusters, where each is represented as a *mean*, i.e. $\Theta = [\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k]$. The means, $\boldsymbol{\mu}_i$ can be thought of as a linear combination of the typical descriptors that occur in the different latent patterns and are commonly referred to as the cluster *centroids* or *centres* and we use the terms interchangeably. The centroids are N-dimensional vectors but they are not necessarily training samples; they are the (mean) average of the training data samples assigned to the centroid. The algorithm works by minimizing inertia, which can be thought of as minimising the within-cluster sum-of-squares error between the training samples and the k centroids. Inertia can be thought of as a measure of how internally coherent the clusters are, i.e. for a given training set $C = [\mathbf{h}_1, \ldots, \mathbf{h}_M]$, it computes:

$$\sum_{i=0}^{m} \min_{j \leq k} (||\mathbf{h}_i - \boldsymbol{\mu}_j||_2),$$

where, given $p \geq 1$, the p-norm of an activity histogram vector is defined as:

$$\|\mathbf{h}\|_p = \left(\sum_{j=1}^{n} |h_j|^p \right)^{1/p}. \tag{5.1}$$

The k-means algorithm has three main steps. First it randomly initialises k centroids from the training samples, then loops between steps: 1) assigning each sample to the closest centroid; and 2) updating the centroids by replacing them with the mean value of all of the samples assigned to each centroid. The difference between the previous and new centroids is computed and the algorithm repeats until this update value is less than a specified threshold δ, i.e. convergence is assumed when the centroids do not move significantly. The basic k-means algorithm is shown in Algorithm 1. The inputs to the algorithm are: a term-frequency matrix, where the rows represent training sample activity histograms, $C = [\mathbf{h}_1, \ldots, \mathbf{h}_M]$, a specified number of clusters k and a convergence threshold δ. It outputs: k centroids, each as an N-dimensional vector, $\Theta = [\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k]$, and a length M vector of cluster assignment labels $L = [l_1, \ldots, l_M]$. The algorithm also maintains the set of data points assigned to the i^{th} cluster L_i, and the previous iteration centroids Θ'.

There are a number of unsupervised techniques to determine the value of k, i.e. the best number of learned cluster centres given the data. One common approach is Schwarz's Bayesian Information Criterion (BIC) [Schwarz et al., 1978]. However, we employ the Silhouette Coefficient (SC) [Rousseeuw, 1987], which we found works well for our task and seems to be more intuitive. This technique automatically determines k that generates the best model for the data. Given a value of k, the SC algorithm computes the following for each training sample $\mathbf{h}_i \in C$: 1) $a(i)$, the average distance (or dissimilarity) between $\mathbf{h}_i$ and all other samples assigned to the same cluster; known as the inter-cluster distance. The smaller the value of $a(i)$, the better the

Algorithm 1 Basic k-means algorithm

procedure k-MEANS ALGORITHM
Input: Activity histograms $C = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_M]$, number of clusters k, threshold δ
Output: Centroids $\Theta = [\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k]$, label assignments $L = [l_1, \ldots, l_M]$
Initialise:
    foreach $\boldsymbol{\mu}_j \in \Theta$ **do**
        $\boldsymbol{\mu}_j \leftarrow \mathbf{h}_i \in C$ (random selection)
    foreach $\mathbf{h}_i \in C$ **do**
        $l_i \leftarrow argmin_{j \leq k} ||\mathbf{h}_i - \boldsymbol{\mu}_j||_2$ (assignment)
Repeat:
    $\Theta' = \Theta$
    foreach $\boldsymbol{\mu}_j \in \Theta$ **do**
        $L_i = \{\mathbf{h}_i \in C : l_i = j\}$
        $\boldsymbol{\mu}_j = \frac{1}{|L_i|} \sum_{L_i} \mathbf{h}_i$ (centroid update)
    foreach $\mathbf{h}_i \in C$ **do**
        $l_i \leftarrow argmin_{j \leq k} ||\mathbf{h}_i - \boldsymbol{\mu}_j||_2$ (assignment)
Until:
    $||\Theta' - \Theta|| \leq \delta$

sample fits its assigned cluster. 2) $b(i)$, the lowest average distance between $\mathbf{h}_i$ and all other clusters that $\mathbf{h}_i$ is not assigned; known as the mean nearest-cluster distance. The cluster with the lowest average distance is said to be the cluster's "neighbour" of $\mathbf{h}_i$ because it is the next best fit for that sample. The silhouette of a training sample is then defined as:

$$s(i) = \frac{(b(i) - a(i))}{\max\{a(i), b(i)\}}.$$

The average $s(i)$ value, over all M training samples, falls within the range of $[-1, 1]$, where higher values are an indication of better, non-over-fitted, models. As such, with SC we avoid the model over-fitting the training samples.

Once the model is trained on the encoded term-frequency matrix, classification of new observations is efficient by calculating the (Euclidean) distance between the new encoded observation and the k centroids. Therefore it does not require maintaining the previous training samples.

5.2.2 Normalising Activity Histograms

Given an $M \times N$ term-frequency matrix C, the constituent activity histograms can have a high dimensionality, i.e. the length of the code book N corresponds to a large number of unique discrete descriptors extracted from the observations. Using k-means, we desire that any two activity histograms with a similar relative proportion of code words to be considered "close" in this high dimensional space, regardless of the absolute frequencies. One technique to achieve this is to unit-normalise each histogram. For example, if we double the frequency of every bin of an activity histogram, after normalisation the two resulting histograms are the same. This is a desirable property since different length observations can have a different number of QSR

intervals and therefore be represented with larger counts of code words (which tend to be in proportion for similar observations). It is worth noting that an observation of shorter temporal length, i.e. in terms of number of timepoints (or poses) does not imply fewer QSR intervals. The exact number of intervals depends upon the number of qualitative spatial relation changes encoded in the interval representation between encoded entities.

Calculating the unit-normal histograms requires dividing each sample by the Euclidean length of the vector. For each $\mathbf{h}_i \in C$:

$$\mathbf{h}'_i = \frac{\mathbf{h}_i}{||\mathbf{h}_i||_1},$$

where the 1-norm of an activity histogram vector is defined in Equation 5.2.1:

5.2.3 k-means Limitations

Simple unsupervised clustering such as k-means is a basic learning method employed by the robot to learn repeated patterns. Using Occam's razor as a heuristic, it suggests to give precedence to simple models: "of two competing theories, the simpler explanation of an entity is to be preferred". For this reason, we implement the simple k-means algorithm. The benefits of using a simple unsupervised clustering include: the algorithm is efficient to run on large training sets; the output is easily interpretable as linear combinations of descriptors (code words); and the classification is computationally cheap. However, there are multiple limitations to using such a simple method; we discuss them next and propose techniques to improve upon them in the sections.

Convergence Criterion

Given enough time to do so, the k-means algorithm is guaranteed to converge Bottou et al. [1995]. However, it may converge to a local minimum, and not to a more useful global minimum. This depends very much upon the random initialization of the centroids in the algorithm. To combat this stochastic process, the algorithm is often run several times with different initializations to try and resolve this potential problem, i.e. to have the best chance of learning the global minimum.

Secondly, the within-cluster sum of squares criterion, which is used to deterministically fit the data samples to the centroids, suffers from the following two issues:

1. It makes the assumption that clusters are convex, which is not always the case. This effects the method's ability to model elongated clusters properly, or manifolds and irregular shapes within certain dimensions. There is no guarantee the underlying activity classes observed by the robot are convex with respect to the encoded discrete descriptors in the N-dimensional feature space.

2. Inertia, or sum of squares error, is not a normalized metric, i.e. we only know that lower values are better and zero is optimal. However, in a high dimensional space, Euclidean distances tend to become inflated (this is an instance of the so-called "curse of dimensionality" problem), i.e. everything becomes far away from everything else.

To alleviate these two issues, we propose dimensionality reduction and a low-rank approximation technique, in the next Section 5.3, in order to map the term-frequency matrix into a lower dimensional feature space whilst maintaining as much of the discriminative information as possible.

Cluster Assignment

Another limitation of the simple unsupervised clustering using k-means is that the training samples are deterministically assigned to only a single centroid, i.e. the closest centroid is selected to represent the sample (based upon Euclidean distance). This relates to each observation being considered as only a member of one activity class. This assumption is not ideal since we know that human behaviours and activities are often compounded. Restricting each observation to just one activity class means that multiple consecutive interactions within the same observation would be considered as the same centroid, when ideally, multiple different activities should be learned. We propose a probabilistic, generative method that learns a probability mixture over the activity classes for each training sample in Section 5.4. This more intuitively describes each observation as a distribution of learned classes, allowing multiple to occur or overlap.

Batch Process

Finally, the process of learning k representative means (using the k-means algorithm) to observational data is an efficient process, however, it is limited by the need to iterate over all the activity histograms in order to perform a single update of the centroids, and multiple updates in order to converge. This can be seen in line 14 in Algorithm 1. This is often referred to as a "batch learning process", i.e. the robot must encode all observations into a term-frequency matrix before learning k centroids given the "batch" of data. Given a requirement for the robot to update its learned centroids with newly observed samples, it would require maintaining each of the previous training samples in memory and repeating the learning process in multiple, ever-increasing batches. This is an expensive and inefficient method to perform continuous, or life-long learning. We propose a probabilistic and incremental learning processes to overcome this limitation in Section 7.2.

5.3 Low-Rank Approximations for Clustering

In order to resolve some of the limitations present with simple unsupervised clustering applied to high-dimensional activity histograms, we draw upon similarities to information retrieval

techniques. That is, we aim to learn a lower dimensional representation of the encoded term-frequency matrix by finding redundancy within the set of descriptors. Redundant descriptors are those that are not useful to separate the training samples into disjoint classes. One reason this might occur is because common sequences of qualitative spatial relations might occur in most training samples, or a sequence of relations might appear very infrequently, possibly due to noisy sensor readings. The most discriminative descriptors are those that contain the most variation within the columns of the term-frequency matrix C, and the assumption is that these are the best descriptors to use to learn human activity classes.

The process is performed using linear algebra techniques which compute linear combinations of descriptors to create new composite descriptors that contain as much variation as possible. By sorting the new descriptors by their ability to discriminate the observations, the most redundant are removed to leave a low dimensional representation in order to recover latent classes encoded in the data. One such technique for this is Latent Semantic Analysis (LSA) [Deerwester et al., 1990], which was originally developed for understanding large corpora of text documents. Given a term-frequency matrix C, the algorithm comprises of two stages:

1. First, compute and apply a specific weighting to each descriptor based upon its variation in the training samples, with the assumption that the most descriptive features have the largest variation.

2. Secondly, a matrix decomposition technique known as Singular Value Decomposition (SVD) is performed. This recovers a collection of eigenvalues and eigenvectors which can be manipulated in order to find the closest low dimensional approximation of the encoded term-frequency matrix.

This section is organised into these two stages; first we describe the weighting applied to each descriptor and secondly, the SVD process. We conclude by highlighting the main limitations that persist when using it to learn human activity classes from a mobile robot.

5.3.1 Term Frequency - Inverse Document Frequency

For each activity histogram, each frequency count of a descriptor, as encoded in an $(M \times N)$ term-frequency matrix C, has its value weighted with respect to the frequency of that descriptor in the entire corpus of observations (C). Here, M is the number of activity histograms (rows) and N is the number of descriptors (columns), i.e. the number of unique code words in the learned code book. The technique to weight each descriptor based upon its frequency is called *term frequency-inverse document frequency* (tf-idf). The tf-idf weighting increases proportionally to the number of times a descriptor appears in an activity histogram and is inversely proportional to the frequency of the descriptor in the entire corpus. That is, it is a measure of how much information observing a descriptor provides.

The weighting is calculated for each code word (descriptor) separately by computing the product of two statistics: 1) *term-frequency*, (tf), i.e. the number of activity histograms that

contain the code word; and 2) *inverse document frequency* (idf) obtained by dividing the total number of activity histograms by the number of histograms containing the code word, and taking the logarithm. Therefore, this process scales the encoded descriptor based upon the fact that some descriptors appear much more frequently in general than others. For example, the sequence of QSR intervals obtained between a person's body joints when they are observed in a "resting position" will most likely appear in the majority of human observations in the corpus. Thus the occurrence of this sequence of QSRs is not discriminative to learn about different human activities and therefore this feature will be given a low weighting by the tf-idf scaling.

To compute the tf-idf scores, we use a Boolean term-frequency (tf) weighting and a logged inverse document frequency (idf) weighting:

$$tf(w_j,\ \boldsymbol{h}_i) = \begin{cases} 1 & \text{if } w_j \text{ occurs in } \boldsymbol{h}_i, \\ 0 & \text{otherwise,} \end{cases}$$

$$idf(w_j,\ C) = log \frac{M}{|\{\boldsymbol{h}_i \in C : w_j \in \boldsymbol{h}_i\}|},$$

where w_j is a unique code word in the code book $\mathcal{V}_{\mathcal{D}}$, $\mathbf{h}_i \in C$ is an activity histogram (one row in the term-frequency matrix C), and M is the total number of activity histograms in the corpus of observations. Finally, the tf-idf weights are calculated as:

$$tf\ idf(w_j,\ \boldsymbol{h}_i,\ C) = tf(w_j,\ \boldsymbol{h}_i).idf(w_j,\ C).$$

It can be seen that if a code word appears in every observation and therefore every activity histogram, its tf-idf weight is equal to $log(\frac{M}{M}) = 0$. Conversely, if a code word appears in only a single activity histogram, its weighting is $log(M)$.

5.3.2 Singular Value Decomposition

Matrix Rank

The goal of Latent Semantic Analysis (LSA) is to find a lower dimensional representation of the term-frequency matrix whilst maintaining as much information as possible. To do this, we recall that the *rank* of a matrix is defined as either: a) the maximum number of linearly independent column vectors in a matrix; or b) the maximum number of linearly independent row vectors in a matrix. For an $(M \times N)$ matrix C, the number of non-zero eigenvalues or singular values is bounded by the rank of the matrix, i.e. at most $min(M, N)$. The rank is equal to r if and only if there exists an invertible $M \times M$ matrix A and an invertible $N \times N$ matrix B, such that:

$$A_{(M\times M)}C_{(M\times N)}B_{(N\times N)} = \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix},$$

where I_r denotes an $(r \times r)$ identity matrix. The rank is commonly computed using the row echelon form of the matrix, since the maximum number of linearly independent vectors in a matrix is equal to the number of non-zero rows in its row echelon matrix [Manning et al., 2008].

Low-Rank Approximation

Given a corpus of observations as represented by an $(M \times N)$ tf-idf weighted term-frequency matrix C, the aim is to recover a matrix which represents a closest approximate matrix to C but with lower rank, i.e. C_r, where $r << min(M, N)$. Finding a lower dimensional representation of a matrix is equivalent to computing a *low-rank approximation*. We can do this by finding a second matrix C_r, requiring it to be as similar as possible to the original matrix C defined by the *Frobenius norm* between two matrices, but with much lower rank, namely r. That is, to miminise the Frobenius norm of the matrix difference $X = C - C_r$, defined to be:

$$||X||_F = \sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N}(X_{ij})^2}.$$

To find a low-rank approximation matrix C_r, we use SVD to compute the eigenvalues and eigenvectors of C. This process provides linear combinations of the descriptors that are used to compute the closest approximate matrix, C_r (with rank r). That is, the most discriminative r linear combinations of descriptors are maintained and the least discriminative are removed resulting in a rank r matrix approximation.

A geometric interpretation is that the eigenvalues represent the scaling values of the matrix in each specific new orthogonal dimension, whereas the singular vectors represent the rotations around those axes. Low eigenvalues are given to vectors which are linear combinations of other vectors, and are therefore not providing extra discriminatory information to the term-frequency matrix. For this reason, LSA helps alleviating the effects of multiple descriptors (or code words) being synonymous which can often be the case when using a sparse representation encoded from observations of humans.

SVD Language

Given an $(M \times N)$ tf-idf-weighted term-frequency matrix C, SVD performs the matrix decomposition:

$$C = U_{(M\times M)}\Sigma_{(M\times N)}V^T_{(N\times N)},$$

where U and V are orthogonal matrices comprising of the singular vectors of C, whilst Σ is a non-increasing diagonal matrix containing the squared eigenvalues of C, i.e.:

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & & \\ 0 & \sigma_2 & \ddots & \\ & \ddots & \ddots & 0 \\ & & 0 & \sigma_r \end{bmatrix},$$

where $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_r \geq 0$ are the *singular values* of C [2]. An orthogonal matrix is a square matrix whose columns and rows are orthogonal unit vectors, i.e. perpendicular to each other:

$$Q^T Q = Q Q^T = I.$$

If we consider the term-frequency matrix C as a linear transformation taking a vector $\mathbf{v}_1$ in its row space to a vector $\mathbf{u}_1 = C\mathbf{v}_1$ in its column space. SVD can be interpreted as finding the orthogonal bases for the row space that gets transformed into an orthogonal basis for the column space, i.e. $C\mathbf{v}_i = \sigma_i \mathbf{u}_i$ for $i \leq r$. The problem translates into finding a set of r orthonormal basis (orthogonal and normal) $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_r$ in which to represent the encoding activity histograms in, i.e. the rows of C, and for which:

$$\begin{aligned} C[\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_r] &= [\sigma_1 \mathbf{u}_1, \sigma_2 \mathbf{u}_2, \ldots, \sigma_r \mathbf{u}_r] \\ &= [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_r] \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_r \end{bmatrix}, \end{aligned}$$

where $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_r$ are orthonormal bases in which to represent the code word vectors, i.e. the column space of C.

An implementation of the Randomized SVD procedure in [Halko et al., 2011] is used and requires only $2(q+1)$ passes over the matrix (for $q = 1$ or 2), making it an efficient method for large matrices.

Interpreting SVD for Activity Histograms

Our aim is to only recover a relatively small number of latent concepts r from the encoded term-frequency matrix, where $r << M$ or N. The assumption is that human activities are repeated multiple times within the robot's human observations, and that the most repeated and coherent patterns of qualitative spatial relations relate to concepts which can be extracted

[2] Here we represent Σ as the $(r \times r)$ square matrix, but it is actually an $(M \times N)$ matrix containing r non-zero singular values on the diagonal, and zeros everywhere else.

using SVD.

Examining the decomposition, the non-zero eigenvalues in the diagonal matrix Σ represent each of the most discriminative new compositional features, known as concepts. These latent concepts can be thought of as the activity classes encoded in the original term-frequency matrix. The columns of the left singular $(M \times M)$ matrix U contain the eigenvectors of CC^T, since $CC^T = U\Sigma\Sigma^T U^T$, and holds information about which observations are assigned to which latent activity class (concept). The columns of the right singular $(N \times N)$ matrix V contain the eigenvectors of $C^T C$, since $C^T C = V\Sigma^T \Sigma V^T$, and specify the linear combination weights for each descriptor (code word) used to describe each concept.

SVD provides an efficient method to recover linear combinations of encoded code words in order to learn the most discriminative latent concepts embedded in the term-frequency matrix, and a vocabulary with which to describe them in an unsupervised setting.

5.3.3 Low-Rank Approximation Limitations

Concept Assignment

When compared to simple unsupervised clustering, LSA provides a more detailed classification of each of the training samples observed, i.e. as a linear combination of the latent concepts. This is a marked improvement on a deterministic assignment, since human observations are often varied and do not belong to a single activity class.

Given the matrix decomposition, i.e. the left/right singular matrices describe the linear combinations of observations to concepts U, and code words to concepts V; one limitation is that both U and V are orthogonal matrices. The implication of the orthogonal matrices is that the concepts extracted cannot share original descriptors, e.g. a specific code word cannot be significant in two separate concepts. This is a limitation of the LSA technique since ideally, we would like to be able to represent all human activities and an assumption is that some activities share qualitative spatial features. A solution to this issue is proposed by using a generative probabilistic model in Section 5.4.1.

Batch Process

Other limitations, similar to those encountered when using simple unsupervised clustering, still persist. That is, LSA is also a batch learning algorithm which requires the entire term-frequency matrix C to be encoded before the training process can occur. A slight improvement is the number of iterations the algorithm requires to learn the eigenvalues $2(q+1)$, which is likely to be fewer than the k-means algorithm. However, like k-means, to incorporate new observations into the training, the learning must be re-performed with a larger term-frequency matrix. New observations can be represented by their similarity to already learned concepts, but they cannot contribute to the model and affect the concepts.

Selecting the Rank r

Selecting the best number of eigenvalues and eigenvectors to represent the new low-rank approximate matrix C_r is often challenging and akin to selecting the number of centroids k, in the unsupervised k-means clustering method. One technique for selecting a good value of r is to plot the variation of each feature, as given by the eigenvalues, on a 2D axis defined as a *scree plot* [Cattell, 1966]. Since the eigenvalues recovered in Σ are non-increasing, the ideal pattern in the scree plot is a steep curve, followed by a bend, often called the "elbow point", followed by a more flat/horizontal line indicating the new descriptors that add little or no variance. Selecting the bend or elbow point is equivalent to setting a lower bound on the change in variation between any two consecutive eigenvalues. This technique allows a good value of r to be ascertained, however, the exact number can often depend upon the task. The scree plot gives an understanding about how much variance is lost when different values of r are selected and different low-rank approximate matrices are computed. This technique is used in the Experiments in Section 6.1.

5.4 Learning LDA Topic Distributions of Activities

In this section we introduce an information retrieval technique in order to resolve some of the limitations when using unsupervised standard clustering or low-rank approximation techniques to learn human activities. The main intuition is that an observation of a human should be modelled in such a way that allows for multiple, overlapping classes of behaviour or activity to occur. For this reason we introduce Latent Dirichlet Allocation (LDA) which is commonly referred to as Topic Modelling. Topic modelling can be considered a case study in probabilistic modelling and touches upon many aspects including: directed graphical models, conjugate priors, time series modelling, approximate posterior inference, mixed membership models and others. The key idea is two fold:

1. a *topic* is defined as a multinomial distribution over a vocabulary of code words and describes a particular thematic structure present in a corpus.

2. a *document* or activity histogram is represented as a probabilistic mixture over topics which is known as the *proportions* or *mixing* vector.

An assumption is that similar documents or activity histograms use a similar group of code words, and therefore the co-occurrence of code words can be used to identify the topics. This framework allows for each observation of a human to be modelled as a mixture of activity classes occurring, and to simultaneously recover the latent activity classes. Resulting in a low dimensional representation of each observation and of the activity classes themselves.

In this section we first introduce LDA and describe the generative probabilistic model that is assumed to have generated our observational data. We describe the LDA graphical model

and define the joint probability distribution (JPD), plus the approximate inference techniques which are used to estimate the posterior, such as Collapsed Gibbs Sampling [Geman and Geman, 1984, Casella and George, 1992]. Lastly, we discuss techniques that can be used to visualise and better understand the output of a topic model.

5.4.1 Latent Dirichlet Allocation

A topic model is a generative probabilistic model of a collection of documents or bags-of-words, comprised of discrete units. Commonly, the collection of discrete data modelled represents the frequencies of natural language words extracted from text documents, and the main emergent themes or concepts present in the documents are learned and referred to as *topics*. For example, given a corpus of webpage documents, latent topics inferred could include general themes such as finance, sport or weather. Each topic is represented by a distribution over a *vocabulary* of code words, with high probabilities given to words that most represent the topic, e.g. "money" or "bank" in the finance topic. For each document a distribution of topics known as a *proportions vector* is learned that represents the mixture of topics exhibited in the specific document. For example, a document could be represented as 70% relating to finance and 30% to sport, implying that 70% of the words in the document were generated from the finance topic distribution, which in turn gives higher probability to finance-like words.

Prior Distribution

One can express a belief about an uncertain quantity before any evidence is observed by proposing a prior probability distribution. A key aspect of LDA is that it introduces two prior distributions, 1) on the mixture of topic proportions, i.e. $\boldsymbol{\theta} \sim Dir(\boldsymbol{\alpha})$ and 2) on the topic distributions themselves, i.e. $\boldsymbol{\phi} \sim Dir(\boldsymbol{\beta})$. These prior distributions provide some expert knowledge to alter the Dirichlet probability distribution parameter, and change the simplex (the distribution over distributions) that the samples drawn exist on.

The Dirichlet distribution is an exponential family distribution over the simplex, i.e. positive vectors that sum to one. It is a distribution over distributions, and in this case, it is a distribution over multinominal distributions. Given T topics, the density of a T-dimensional Dirichlet distribution, parametrised by $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_T]$, over the T-dimensional multinomial distribution $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_T]$ lies on the $T-1$ simplex and is given by:

$$p(\boldsymbol{\theta}|\boldsymbol{\alpha}) = \frac{\Gamma(\sum_{i=1}^{T} \alpha_i)}{\prod_{i=1}^{T} \Gamma(\alpha_i)} \prod_{i=1}^{T} \theta_i^{\alpha_i - 1}, \tag{5.2}$$

where Γ is the gamma function and the sum of the proportions vector for a document is one, i.e. $\sum_{i=1}^{T} \theta_i = 1$ and where $\theta_i > 0$ for all $i \in T$. Note, usually in LDA a symmetric Dirichlet distribution is used, i.e. $\alpha_1 = \cdots = \alpha_T = \alpha$. The Gamma function is defined in Appendix C.3.

Dirichlet Simplex

The prior Dirichlet distribution on $\boldsymbol{\theta}$ allows for a more uniform topic proportions vector to be sampled, with the amount of variance depending on the value of the hyperparameter α. Recall that $\boldsymbol{\theta} \sim Dir(\boldsymbol{\alpha})$ and $\boldsymbol{\phi} \sim Dir(\boldsymbol{\beta})$. We continue by discussing the effect of the hyperpameter on the per-document proportions vector $\boldsymbol{\theta}$, however, the same applies to the per-topic distribution $\boldsymbol{\phi}$. For higher values of α_i, the greater the amount of the total distribution "mass" is assigned to θ_i (recall that $\sum_{i=1}^{T} \theta_i = 1$). If all α_i are equal, the distribution is symmetric. If $\alpha_i < 1$, it can be thought as pushing away θ_i toward the extremes. While, when α_i is high it attracts θ_i towards topic i. If $\alpha_1 = \cdots = \alpha_T = 1$, then the points are uniformly distributed.

In LDA, a symmetric Dirichlet prior is often used, so no topic proportion θ_i has preference over the others. This is achieved by using a single hyperparameter α, where $\alpha_1 = \cdots = \alpha_T = \alpha$. If α is less than one, the resulting simplex will be "bowl" shaped, meaning that the multinomial samples drawn will all be far from the centre and on the edges of the simplex. This will favour a small number of topics per-document, and gives very low mixing proportions to the other topics. This is an assumption that each document is comprised of only a small number of topics. Figure 5.2 shows an example of a symmetric Dirichlet distribution with three values of α, (left:) ≤ 1. (centre:) 4 and (right:) 2. Sampling a topic distribution from the Dirichlet prior distribution can be thought of as selecting a multinomial distributions which lives on this simplex. For example, a sample from the distribution when $\alpha = 4$, the multinomial drawn is much more likely to be uniform across the topics, i.e. a mixture of all three topics only varying slightly between the three, whereas, when $\alpha \leq 1$, the multinomial sample is much more likely to represent only one or two of the topics, as there is not as much "mass" in the centre of the simplex. Justification into the choice of the hyperparameters are given in the experimental sections.

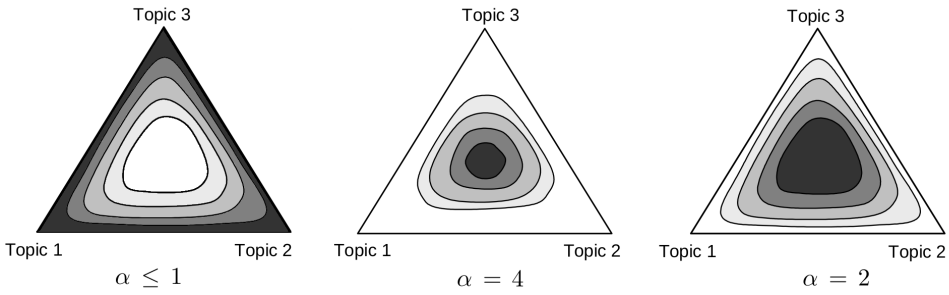


Figure 5.2: Illustrating the symmetric Dirichlet distribution for three topics on a two-dimensional simplex. Darker colours indicate higher probability. Left: $\alpha \leq 1$. Centre: $\alpha = 4$. Right: $\alpha = 2$. When $\alpha \leq 1$, the simplex is bowl-shaped. As α increases, the simplex becomes more peaked around centre.

Activity Analysis Language

We employ the use of LDA to model the robot's observations, where each is encoded as a bag-of-qualitative-words akin to a document in the text analysis setting. Here, a set of topics is learned from the encoded observation data, each characterised as a multinomial distribution over a vocabulary of code words; representing the thematic structure in the abstracted observations. These latent topic distributions are considered as describing the common human activity classes. Each observation of a human is analogous to a document and modelled as a random mixture over these latent topics, represented as a proportions vector. This can be thought of a low-dimensional representation of the human observation over the set of learned activity classes.

We summarise the main human activity analysis language; accumulating concepts from the previous chapter and introducing the corresponding LDA language:

- a *bag-of-qualitative-words* is a bag of observed qualitative descriptors, $d = \{w^1, w^2, \dots\}$, where each w^i represents a temporally connected sequence of qualitative spatial relations between entities extracted from a human observation. This is akin to a document encoded as a bag-of-words in text analysis (however, our graph path-words maintain local temporal structure).
- a *video corpus* is a collection of M observations, where each human detected is recorded by the robot.
- a *code word*, w_i, is a basic unit of activity as encoded in our representation. The set of unique qualitative descriptors (graph paths) forms a discrete vocabulary defined as a code book $\mathcal{V}_D$.
- an *activity histogram* $\mathbf{h}$ is a vector of occurrence counts over the N-dimensional code book $\mathcal{V}_D$ and encodes one bag-of-qualitative-words. Encoding the entire corpus results in an $(M \times N)$ term-frequency matrix of constituent activity histograms, $C = [\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_M]$, corresponding to M bags-of-qualitative-words, i.e. $D = \{d_1, d_2, \dots, d_M\}$.
- a *topic* $\boldsymbol{\phi}$, is a multinomial distribution over a vocabulary code book $\mathcal{V}_D$, giving each unique code word a probability of being observed conditioned on that topic.
- a *topic proportions vector* $\boldsymbol{\theta}$, is a multinomial distribution that characterises the mixing proportions of each topic present in a bag-of-qualitative-words d.

A bag-of-qualitative-words representation, which encodes occurrences graph paths in an unordered "bag". In probability theory, this relates to an assumption of *exchangeability* of the code words in an observation. Less commonly stated, there is also an assumption that the documents are exchangeable within a video corpus. A classic representation theorem establishes that any collection of exchangeable random variables has a representation as a mixture distribution [de Finetti, 1990]. Thus, Latent Dirichet Allocation results in probabilistic mixture distributions of code words.

Generative LDA Model of Human Activity

LDA for human activities is a probabilistic generative model that generates a collection of activity histograms (or bags-of-qualitative-words) from a set of underlying topics. Figure 5.3 shows the intuition behind the LDA generative process for an encoded human observation. For each activity histogram, or bag-of-qualitative-words d, the process is characterised as follows:

1. sample a per-document topic proportions vector, $\boldsymbol{\theta}_d$, from a prior Dirichlet distribution parameterised by α, i.e. $\boldsymbol{\theta}_d \sim Dir(\alpha)$. An example sample drawn over three topics is shown far right as a histogram in Figure 5.3. This represents a multinomial distribution drawn from the Dirichlet simplex. These are the mixing proportions for the activity histogram.

2. for each of the N_d code words in the bag-of-qualitative-words d:

   – draw a per-word topic assignment, $z_{d,n}$, from the proportions vector, i.e. sample an assignment coin $z_{d,n} \sim Multinomial(\boldsymbol{\theta}_d)$. For example, the pink topic in Figure 5.3 is sampled and shown with a pink coin, followed by yellow, then pink again etc. This allows each code word in the bag to be drawn from different topic distributions, respecting the topic mixing proportions in $\boldsymbol{\theta}_d$ and facilitating the mixing of topics within an observation.

   – for each topic assignment, draw a word, $w_{d,n}$, from the multinomial topic distribution conditioned on $z_{d,n}$, i.e. sample a code word $w_{d,n} \sim Multinomial(\boldsymbol{\phi}_{z_{d,n}})$. For example, from the pink topic assignment (coin) the code word (ι'_4 *meets* ι'_5) is drawn, which can be seen far left in Figure 5.3 as the a highly probable word in the pink topic. Then, the word (ι'_1 *meets* ι'_2) is drawn from the yellow topic assignment, etc.

3. this process repeats to generate all M bags-of-qualitative-words.

LDA as a Graphical Model

In reality, the robot only observes the bags-of-qualitative-words and not the mixing proportions vectors or the assignment of each code word into topics. This is the latent structure (variables) of the model that we aim to infer, i.e. p(topic distributions, proportions, assignments | activity histograms). Given a collection of activity histograms, this corresponds to inferring the three sets of latent variables:

– $\Phi = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_T]$ per-corpus topic distributions, where each $\boldsymbol{\phi}_i$ is a distribution over the entire code book $\mathcal{V}_D$,

– $\Theta = [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M]$ per-document topic proportions vectors, where each $\boldsymbol{\theta}_d$ is a distribution over the T topics,

– $\mathbf{z}$ is the assignment of all observed code word tokens to topics, for all activity histograms.

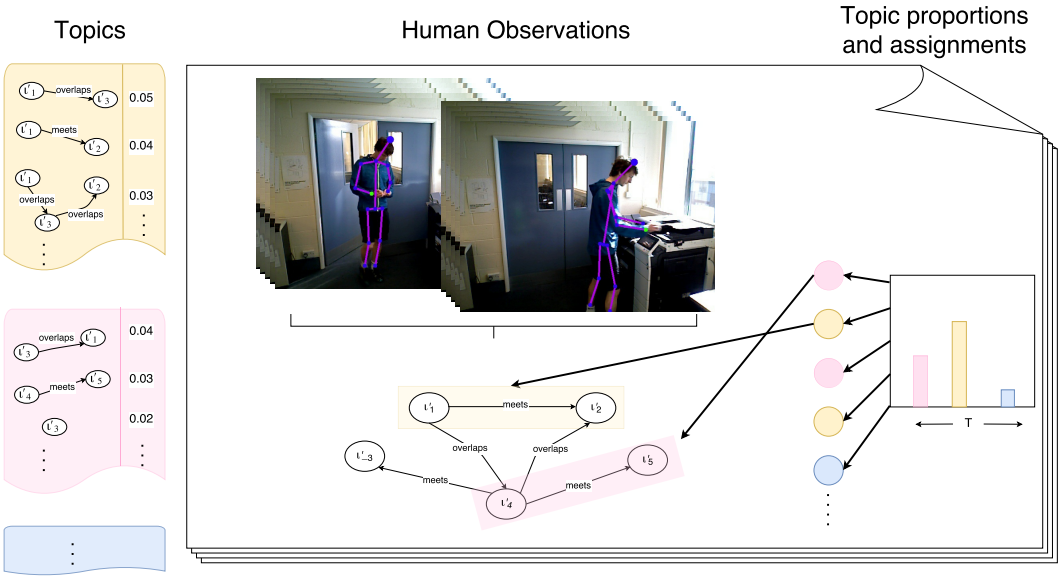
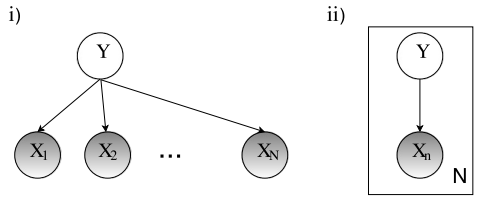


Figure 5.3: Generative LDA model of human activity. (left:) Three topic distributions over the code book (yellow, pink and blue) along with three top-probability words in the yellow and pink topics. (centre:) Generated interval graph and bag-of-qualitative-words obtained from encoding one human observation. (right:) Topic proportions vector (pink, yellow, blue histogram) and word assignments as a column of samples coins drawn (pink coin, yellow, pink, etc.). Best viewed in colour.

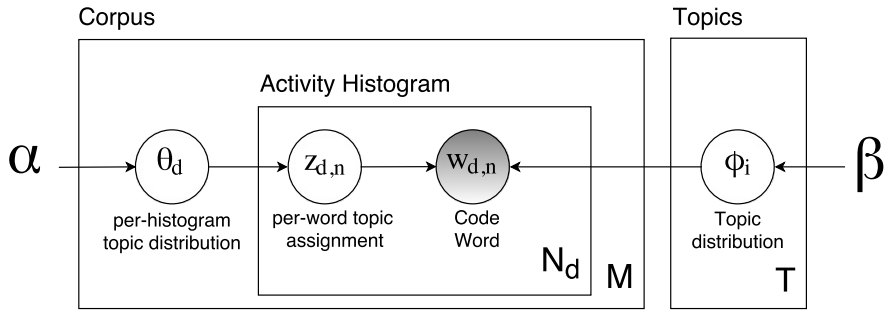
That is, $p(\Phi, \Theta, \mathbf{z}|D)$. To do this, we present LDA as a Bayesian Network or Directed Acyclic Graph (DAG). This is an intuitive way of representing and visualising the relationships that exist between the variables that make up a topic model and corresponds to a specific factorization of the joint probability distribution (JPD).

In a DAG, nodes represent random variables and directed edges between nodes reflect conditional dependencies between variables. A common presentation technique is to depict observed random variables using shaded nodes, and non-shaded nodes for latent variables. Figure 5.4a(i) is an example DAG, showing N observed random variables, $X_1, \ldots, X_N$, conditionally dependent upon a latent variable Y. Plate notation can be useful when dealing with large number of random variables. It is used to highlight replicated random variables, with the number of random variables marked in the lower right corner of the plate. The DAG shown in Figure 5.4a(ii) is equivalent to (i), however, is drawn using plate notation which uses a rectangle *plate* around the repeated random variables. The JPD of this model is:

$$\begin{aligned} p(X_1, X_2, \ldots, X_N, Y) &= p(Y)p(X_1|Y)p(X_2|Y)\ldots p(X_N|Y), \\ &= p(Y)\prod_{n=1}^{N} p(X_n|Y). \end{aligned}$$



(a) (*i*) DAG showing N observed random variables conditionally dependent upon a single latent variable Y. (*ii*) Equivalent DAG using plate notation highlighting the N repeated nodes.



(b) DAG representation of LDA for human activities using plate notation. For activity histogram d: $\boldsymbol{\theta}_d$ (topic proportions vector for histogram d); $w_{d,n}$ (n^{th} observed code word, shaded grey, in histogram d); $z_{d,n}$ (n^{th} code word topic assignment in histogram d); $\boldsymbol{\phi}_i$ (i^{th} topic distribution over code book); α, β (Dirichlet hyperparameters).

Figure 5.4: Bayesian Graphical model representations. Nodes represent random variables, links between nodes are conditional dependencies, plates are replicated components, and shaded nodes are observed random variables.

The DAG representing the LDA model for human activities is shown in Figure 5.4b. The DAG highlights the three-layer hierarchical Bayesian model and uses plate notation to simplify the graph and is considered a three-layer Bayesian model since:

1. the Dirichlet hyperparameters, α and β, are corpus-level parameters.

2. the topic proportions variables, $\boldsymbol{\theta}_d$ for each bag-of-qualitative-words d in the corpus, are activity histogram-level parameters, sampled once per activity histogram.

3. the variables $z_{d,n}$ and $w_{d,n}$ are code word-level and sampled once for each code word in a bag-of-qualitative-words.

Formally, for M observed activity histograms, each containing N_d code words, and a set of T topic distributions, the joint probability distribution over the observed and latent variables

within the DAG define the posterior probability as:

$$p(\boldsymbol{\phi}_{1:T}, \boldsymbol{\theta}_{1:M}, \mathbf{z}, d_{1:M}|\alpha, \beta) = \prod_{i=1}^{T} p(\boldsymbol{\phi}_i|\beta) \prod_{d=1}^{M} p(\boldsymbol{\theta}_d|\alpha) \left(\prod_{n=1}^{N_d} p(z_{d,n}|\boldsymbol{\theta}_d) p(w_{d,n}|\boldsymbol{\phi}_{1:T}, z_{d,n}) \right) \quad (5.3)$$

The three representations (*i*. the intuitive generative process, *ii*. the Bayesian DAG and *iii*. the JPD) are equivalent ways of describing the probabilistic assumptions behind the LDA model. The robot uses posterior expectations to perform inference and learn the latent structure of the LDA model. We introduce the inference problem in the next section.

5.4.2 Approximate Inference

Given a joint probability distribution, like the one defined in Equation 5.3, we can answer possible inference queries by marginalization, i.e. summing out over irrelevant variables. Given the observed activity histograms as encoded in a term-frequency matrix C, inference allows us to estimate the latent variables, the topic distributions and the mixing proportions vectors. This can be considered as finding latent patterns in the data which best separate it into meaningful topics. The aim is to determine the latent topics based on the observed code words in the term-frequency matrix. This translates as computing the posterior distribution of the latent variables given a collection of activity histograms:

$$p(\boldsymbol{\phi}_{1:T}, \boldsymbol{\theta}_{1:M}, \mathbf{z}_{1:M}|d_{1:M}, \alpha, \beta) = \frac{p(\boldsymbol{\phi}_{1:T}, \boldsymbol{\theta}_{1:M}, \mathbf{z}_{1:M}, d_{1:M}|\alpha, \beta)}{p(d_{1:M}|\alpha, \beta)}. \quad (5.4)$$

This posterior distribution is intractable to compute in general. More details about how to marginalise over variables can be found in [Dickey et al., 1987] and [Blei et al., 2003]. However, we use Collapsed Gibbs Sampling [Griffiths and Steyvers, 2004, Lynch, 2007] as an approximate inference technique that is based upon Markov chain Monte Carlo (MCMC) sampling. The key idea is to generate posterior samples from the conditional distribution by iterative sampling using a Markov Chain [Jordan, 1998].

Posterior Estimate of z

Here, we sample to topic assignments, $\mathbf{z}$. The Gibbs Sampling procedure marginalises out the topic distributions and the mixing proportions, which is also know as Rao-Blackwellization [Blackwell, 1947]. The method selects each code word token in the collection in turn and estimates the probability of assigning that current code word token z_i to each topic j, conditioned on the topic assignments to all other code word tokens. This is random at first, but after iterating it converges to sensible assignments. We denote $\mathbf{z}_{-d,i}$ as the code word assignments for all observations, excluding the i^{th} code word in the d^{th} bag-of-qualitative-words. From this conditional distribution, a topic is sampled and stored as the new topic assignment for this code word token. For a given bag-of-qualitative-words, d, the conditional distribution

that a code word token is equal to the j^{th} topic is written:

$$p(z_i = j|\mathbf{z}_{-d,i}, D, \alpha, \beta),$$

which is shown in [Griffiths and Steyvers, 2004] can be calculated using two counts: C^{WT} and C^{DT} with size $(N \times T)$ and $(M \times T)$ respectively (word-topic and document-topic counts). $C^{WT}_{i,j}$ contains the number of times code word i is assigned to topic j, not including the current instance i, and $C^{DT}_{d,j}$ contains the number of times topic j is assigned to some code word token in the bag-of-qualitative-words d, not including the current instance i; these counts are computed from the assignments $\mathbf{z}$. Then:

$$p(z_i = j|\mathbf{z}_{-d,i}, D, \alpha, \beta) \propto \frac{C^{WT}_{i,j} + \beta}{\sum_{i=1}^{N} C^{WT}_{i,j} + \beta} \frac{C^{DT}_{d,j} + \alpha}{\sum_{t=1}^{T} C^{DT}_{d,t} + \alpha} \tag{5.5}$$

The general steps of Collapsed Gibbs Sampling are:

1. randomly initialise all the code words to topics, defined as assignments in $\mathbf{z}$.
2. sample one bag-of-qualitative-words d:
3. for each code word in d:
   - remove the current topic assignment of the code word; we use the notation $\mathbf{z}_{-d,i}$ to define the vector $\mathbf{z}$ excluding the i^{th} assignment in bag d,
   - calculate the topic the current word belongs to, given the current assignment of all the other code words in the bag. That is, using the count matrices C^{WT} and C^{DT},
4. repeat for all code words in the bag.

Posterior Estimate of $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$

The Gibbs Sampling procedure described above gives direct assignments $\mathbf{z}$ for each code word observed, however, we are also interested in the estimate topic distributions and mixing proportions, $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$. These too can be calculated using the count matrices as follows:

$$\phi_{i,j} = \frac{C^{WT}_{i,j} + \beta}{\sum_{k=1}^{N} C^{WT}_{k,j} + \beta}, \qquad \theta_{d,j} = \frac{C^{DT}_{d,j} + \alpha}{\sum_{k=1}^{T} C^{DT}_{d,k} + \alpha}, \tag{5.6}$$

where $\phi_{i,j}$ is the probability of word type i for topic j, and $\theta_{d,j}$ is the proportion of topic j in bag d.

5.4.3 LDA Limitations

The LDA model provides a method to estimate a set of topics to represent the main themes encoded in a term-frequency matrix, where each topic is defined as a distribution over a vo-

cabulary of code words. These topics are then considered as human activity classes, where the highly probable code words are considered discriminative features to understand these human activities. The key advancement over previous techniques proposed is that each human observation is summarised by a multinomial distribution over the set of learned topics, that is, as a mixture of activity classes. This translates as assuming that each observation was generated from multiple activity classes being performed by the person being observed by the robot. This previously was a limitation of both standard clustering and low-rank approximation techniques, where each observation supported a single activity class, and where the input was transformed into linearly independent dimensions. Another benefit of LDA is that expert knowledge about the observations or the topics can be proposed by altering the prior probability distribution on the topic proportions vectors $\boldsymbol{\theta}$, and topic distributions $\boldsymbol{\phi}$.

Topic distributions tend to be sparse vectors over the vocabulary of code words, hence they are often difficult to visualise and interpret. One approach that works well in the literature is to obtain the set of natural language words that have high probability in each topic, and present these; given such a set, it is often easy to conceive the meaning of a specific topic [Chang et al., 2009]. However, this approach relies on semantic knowledge about the set of natural language words, which we can not rely on in our setting. One approach is to plot the distributions of the top-probable words in each topic. An example of this can be seen in Figure 5.5, where three topic distributions are shown, and the highest ≈ 70 code words have been selected.

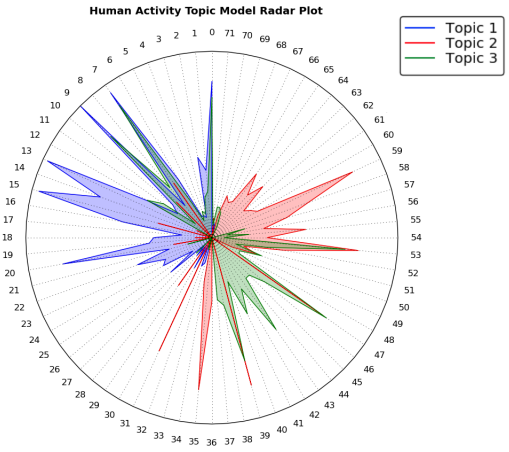


Figure 5.5: Example Radar Plot showing the probability over the top probable code words for three topic multinomial distributions.

However, LDA does not provide a solution to all the limitations. For example, the number of topics must be set in advance in this LDA model. In our experiments section, we use the same low-rank methods of selecting the best number of topics to compute in advance, i.e. finding the number of top eigenvalues in the term-frequency matrix. However, an incremental approach to this problem is also proposed. Also, Collapsed Gibbs Sampling is a batch inference

technique. We propose to use an online variational inference technique to overcome this limitation in Section 7.2.1. Topic models that include time or temporal states can also be modelled with additional random variables [Blei and Lafferty, 2006], however, these rely on observations significantly changing over time, such as the robot changing environments, which is a research direction we did not pursue.

5.5 Concluding Remarks

To summarise this section, we have presented three techniques for unsupervised learning of human activity representations from encoded human observations. That is, we have demonstrated simple unsupervised clustering (k-means algorithm), a low-rank linear algebra approach (LSA) and lastly, a probabilistic generative method (LDA) of learning themes as topics. The three increasingly sophisticated approaches all require the human observations be encoded into a term-frequency matrix that summarises the frequency counts of discrete qualitative descriptors (code words) observed. Each method is able to represent encoded themes in the term-frequency matrix, which we consider as a low-dimensional representation of human activities. Further, given the learned activity classes, each can represent a new observation as either its closest activity, or a linear combination of closest concepts. The LDA model specifically is able to model each observation as a mixture of activity classes occurring, which we believe is required to model real human behaviour observed "in-the-wild", and not staged human activities performed in front of a static camera.

In the next chapter, we demonstrate the capabilities of each learning method on three video datasets, two of which have been recorded from an autonomous mobile robot observing real environments. In Chapter 7 we propose practical methods to remove the remaining limitations, such as, the batch learning process and the temporal segmentation of video clips.

Chapter 6

Experiments: Activity Class Learning

In this chapter we describe our experiments and present empirical results of unsupervised learning of human activity classes from three challenging, real-world datasets of human activity. The chapter is split into two main sections, first we present results on learning human motion behaviours from a six week deployment where the robot recorded only human trajectory data. Secondly, we focus on more detailed 3D human body pose sequences and introduce two datasets which contain such detections: 1) a popular activity dataset recorded using a static camera set-up where 10 activity classes are scripted in advance. The human activities are scripted in advance, but contains dynamic objects tracked in the scene; 2) a real-world dataset recorded from the mobile robot over a period of one week observing in a university student common room and kitchen area.

For human trajectories, we describe the experimental set-up, the filtering techniques used to remove noisy observations and the implementation details of experiments. Given incomplete trajectory observations, we show that the robot is able to learn a set of consistent and meaningful behaviours and patterns using simple unsupervised clustering (k-means clustering). Then, given a new observation, the robot is able to query the most similar motion pattern learned and predict an occupancy grid of future discrete locations. This can enable the robot to assist in a specific task by moving towards or away from the predicted locations depending upon the task. For example during a long-term deployment, the robot's behaviour was altered when an observed trajectory was significantly different (based upon a distance threshold) to the learned motion behaviours. The robot would then approach this detected person and request for them to swipe their security badge. The experiments and results presented in this section appear in [Duckworth et al., 2016b].

The two human body pose datasets allow the robot to learn more detailed human activity behaviours using qualitative relations between each body joint locations and key objects in the

environment. To do this, the robot uses more sophisticated learning methods, such as LSA and LDA which are better suited to this task and improve upon the performance of simple unsupervised clustering. The human activity dataset recorded from a mobile robot is manually annotated by volunteers, and temporally segmented to produce a corpus of video clips each with a single, labelled, human activity occurring. We present the experimental set-up and implementation details of the experiments and show that the human activity classes learned are consistent and align well with human annotated ground truth labels for both datasets. The experiments and results presented in this section appear in [Duckworth et al., 2016a, 2017].

6.1 Learning Human Motion Patterns from Trajectories

Here we present results of learning consistent and meaningful human motion patterns from an autonomous mobile robot's partial human trajectory observations. First we introduce a dataset of human trajectories and the techniques used to filter the incomplete and noisy observations. There is no ground truth with respect to the purpose or intent of a trajectory, so our aim is to evaluate the robot's ability to predict a *target area* on the map plane which is used to predict future trajectory poses. Then, details of the experimental procedure are given, including the environment and static object locations, the qualitative spatial representations used, the graph path and code words parameters, etc.

Given the lack of ground truth information, we evaluate our unsupervised qualitative relational framework in the following ways: 1) We first visualise the human trajectories which are close to the learned k-mean cluster centres in order to qualitatively investigate the emergent motion patterns. This allows for post-processing and to map them to specific human behaviours. 2) Secondly, we assess the consistency of the classification and predicted target area for a new, unseen, trajectory. We evaluate how well the system classifies a newly observed trajectory into one of the unsupervised learned motion behaviours using the mean values of recall, precision and $F1$-score after observing increasing percentages of new trajectories. 3) Finally, we show the motion patterns learned can be used to predict the future target area of a trajectory on the map plane and demonstrate how the performance of these predictions improves with the quantity of observations available to the robot.

6.1.1 Human Trajectory Dataset

Data collection took place during a six week deployment at the UK offices of the G4S security company, using a STRANDS Metralabs Scitos A5 mobile robot[1]. The robot followed a pre-specified schedule during the deployment period which involved patrolling, stationary observation, and object search tasks during weekday working hours (Mon-Fri, 9am-5pm). It was stationary at its charge station during weekends and public holidays. Whilst on its charge

[1]The dataset collected, along with meta-data such as maps and ROS messages, is available at: http://doi.org/10.5518/34.

station, the robot would perform tasks such as database backups and batch learning, which is when the unsupervised human activity learning was performed. Whilst patrolling and performing observational tasks the robot detected and tracked human trajectory poses.

During the six week deployment the robot successfully observed and recorded just over $42,000$ human trajectories and each was stored as a sequence of 2D Cartesian poses in the database, as a trajectory ROS message. Details of the trajectory filtering are given next, but the key idea is to remove any trajectories that are considered as noise (which is not uncommon with a laser based leg-detector). Recall, the laser sensor has a 180 degree field of view, however, this was often occluded by obstacles in the environment and fast moving people were only detected briefly. After filtering, the total number of human trajectories was reduced to $3,981$ with (mean) average length of 2.2m (median: 1.8m and range: 1m$-$6.5m). The short length of trajectories is due to the fact that they are collected using an embedded, mobile robot with limited sensors and this is in contrast to capturing the complete motion of a person in the environment. For example a person could walk down the approximately 25m long corridor but the sensor limitations restrict detecting these poses. The filtered dataset of trajectories can be seen displayed overlaid onto the G4S metric map as in Figure 6.1. The direction of the motion is highlighted by colouring the trajectory poses from blue to increased amounts of red. Trajectory poses outside of the metric map boundaries are due to noisy sensor readings, and are removed in the following analysis.

Obtaining ground truth information relating to the destination or purpose of each trajectory would require the robot to either interrupt and ask the human about their intention, or the use

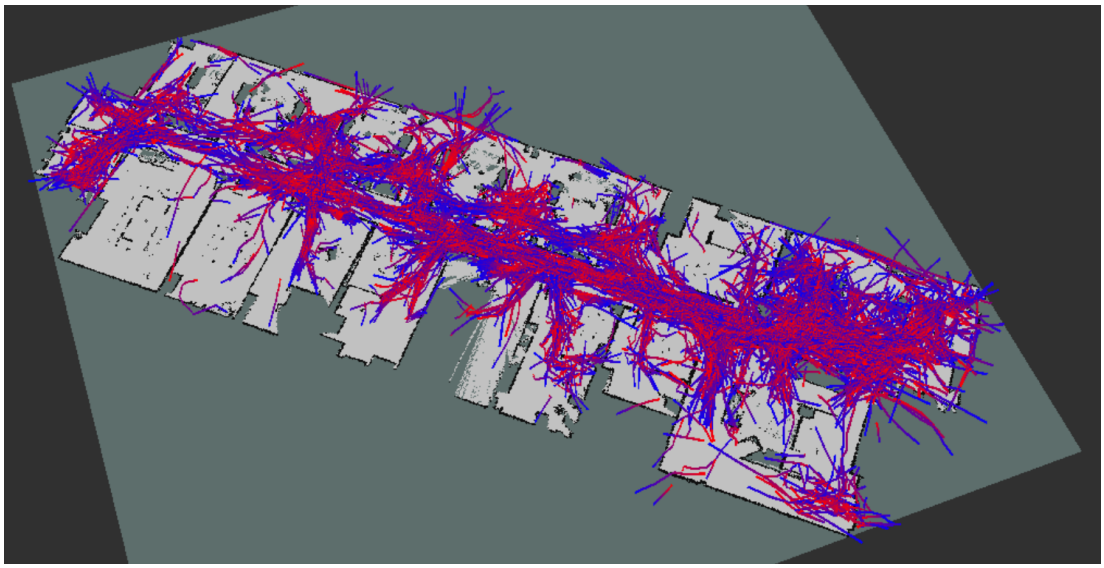


Figure 6.1: Human trajectories dataset consisting of 3,981 trajectories collected during a six week monitoring of human office environment. Colour indicates the direction of motion, blue to red.

of an elaborate motion-capture set-up where intent could be inferred from final destinations. However, neither of these were feasible in our deployment due to the operational conditions of the real-world office environment, and the invasive approach would ultimately disturb the trajectories themselves. Instead, the robot collected observations passively and post-associated learned common motion behaviours to meaningful activities afterwards, which is a further requirement of the unsupervised framework.

The patrolled area was manually segmented into semantic SOMa office-like regions, each with manually annotated key objects, such as desks, bookcases, printers, etc., which can be seen overlaid onto the metric map in Figure 6.2. The deployment was spatially restricted to predominantly observe one specific region during the first week, resulting in much higher number of trajectories in this region in the first week compared to the following five weeks ($1,232$ trajectories in total in this region, 342 in the first week). For this reason we define this region as the *experimental region* which is shown as the left-most semantic region with yellow boundary in Figure 6.2. This region contained a number of employee desks, cabinets, bookcases and tables, along with the robot docking station.

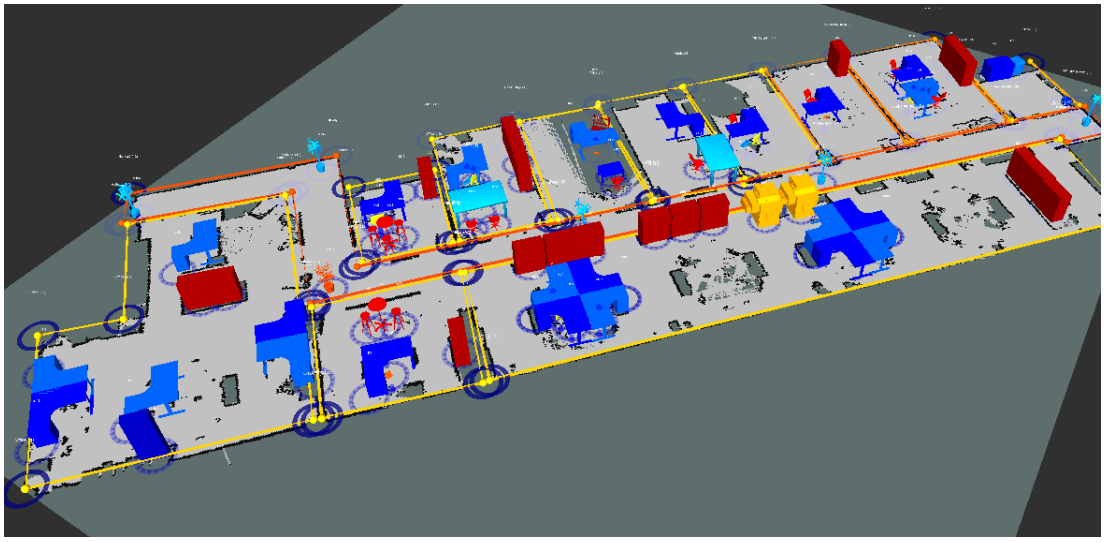


Figure 6.2: G4S metric map, overlaid with SOMa semantic regions (yellow and orange) and object locations (brightly coloured blender models). The experimental region is the left-most SOMa region.

Trajectory Filtering

The motivation for performing filtering on the human trajectories data obtained is the observed discrepancy and variability in the sensor readings. For some sensor readings, it was clear that a trajectory observation did not relate to a detected human in the robot's field of view, but to a static object such as a chair leg, table or large plant. False detections in certain locations were

very common, caused by the obstacles being mis-detected by the implemented leg detector.

Our assumption with the filtering techniques is that static trajectory observations are either noise from sensor discrepancies or they add no additional information about motion behaviours, and in both cases we require methods to filter them out. Therefore, any trajectory with a maximum displacement of less than $1m$ between the first and final pose is considered a stationary object, noisy sensor reading, or not useful for learning motion patterns. Manual inspection of a sub-set of the dataset suggests that some trajectories appear to move very quickly through the map frame, that is, consecutive trajectory poses were far apart. Those trajectories with a displacement per pose greater than a threshold were filtered out and considered as sensory noise.

6.1.2 Implementation Details

The main components of learning motion patterns using unsupervised clustering comprises of two phases: a training phase, and a real-time recognition (or classification) phase. The implementation details of these phases are presented next.

An implementation detail of the Bayes People tracker introduced in Section 3.3.2 is that it groups a number of chronological trajectory poses together into small sub-sequences when it detects a new person. A *sub-sequence* can be thought of as a buffer of $0.4s$ of a human detection which is approximately 10 trajectory poses (given the $25Hz$ sampling rate). The trajectory poses are then stored in the database along with these sub-sequence identifiers allowing for post-analysis of the trajectories. We use these sub-sequence identifiers during the evaluation to mimic the real-time system.

Training Phase

Given a single human trajectory $T = [t_1, t_2, \dots t_i, \dots]$, where each t_i is the detected trajectory pose at timepoint i, with arbitrary number of total poses, the robot computes QSR relations between each trajectory pose and the closest SOMa objects within the same region (in this case, the closest 5 objects). Given the distribution of SOMa objects in the environment, this efficiently captures a good spread of spatial relations throughout the environment without having to limit the set of key objects in advance. For human trajectories we use a combination of QTC to capture relative motion of the person and QDC for relative distance. The QDC thresholds used are $\Delta = [1, 2, 4, 8]$ metres, creating five semantic regions: *touch* [0-1m], *near* (1-2m], *medium* (2-4m], *far* (4-8m] and *ignore* (>8m). These relations are computed in the 2D metric map coordinate frame and for a complete trajectory, results in a sequence of QSR relations for each trajectory-object pair (5 pairs in this case), of QTC-QDC relational pairs, per trajectory pose. For each trajectory-object pair, the sequence of QSR values obtained are then converted into an interval representation and interval graph g, first by compressing repeated relations, and second by abstracting the temporal relationship between each pair of intervals using Allen's Interval

Algebra. All paths through the interval graph are extracted using a maximum path length of three, i.e. $\eta = 3$, and restricting the relations to just one pair of entities, i.e. the trajectory and a single object $\rho = 1$. This process creates a bag-of-qualitative-words $d = \{w^1, w^2, \dots\}$ with length equal to the number of graph paths extracted from the single trajectory observation.

This process is repeated for each of the M trajectories in the training dataset D. Then, a discrete vocabulary is automatically generated as a code book $\mathcal{V}_D = [w_1, w_2, \dots, w_N]$, by taking the union of all the encoded bags-of-qualitative-words resulting in a set of unique graph paths extracted from all interval graphs observed during the training phase; these are considered code words. Each trajectory in the training set is then represented as an activity histogram $\mathbf{h}$ over the code book with the frequency counts of each code word occurring in it. This process creates an $M \times N$ term-frequency matrix C representing the entire training dataset. The unsupervised training phase is concluded with unit normalisation of the activity histograms and clustering into k $N-$dimensional clusters, using the k-means algorithm introduced in Section 5.2.1. Once converged, the algorithm produces a model, Θ, with a set of means $[\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \dots, \boldsymbol{\mu}_k]$, where each $\boldsymbol{\mu}_i$ represents a cluster centroid in the code book space which is considered to represent a typical motion behaviour or pattern present in the training data. We determine the value of k in each training phase using the Silhouette Coefficient (SC).

Recognition Phase

For a newly detected trajectory, the aim is two fold: 1) to recognise the motion behaviour as quickly as possible by accumulating trajectory poses and continually comparing to the previously learned clustering model, Θ, in order to continually predict its most similar motion behaviour. 2) Using this recognised motion behaviour to predict a *target area* on the map plane in order to make a prediction about where the person might be going. We evaluate the target predicted area for the incremental sub-sequences of poses by comparing it with the trajectory's future poses, that are not yet observed. In both cases, the classification process is real-time and bound by the availability of new trajectory poses. The idea is that the robot can generalise even the shortest trajectories and make a prediction of future motion, and that this prediction improves as more of the trajectory is observed.

Classification: Once the clustering model Θ is trained and a new trajectory is being observed, the trajectory poses are available incrementally (split by their sub-sequences) for the duration of time the human is within sensor range. Given the first sub-sequence, $\approx$10 poses, it is encoded using the same steps to obtain a qualitative representation and an interval graph then as an activity histogram over the code book. The histogram is unit-normalised and classified into its closest cluster centroid based upon Euclidean distance. This allows the robot to abstract the first sub-sequence of poses in order to make an initial classification of what behaviour is being observed using only the initial 0.4 seconds of a new trajectory. The classification processes is then repeated for each new sub-sequence of poses, increasing the length of

the trajectory each time more poses are available.

Prediction: Let L_i be the set of trajectories that are assigned to the i^{th} cluster based upon Euclidean distance. We then interpret the i^{th} cluster centre $\boldsymbol{\mu}_i$ as the mean histogram of L_i, i.e. for each activity histogram of a trajectory in L_i, $\boldsymbol{\mu}_i$ represents the 'influence' of each code word in the i^{th} learned motion behaviour. When a new trajectory is observed and classified into the i^{th} motion, we use the cluster centre values in $\boldsymbol{\mu}_i$ to extrapolate the trajectory and make a prediction of what qualitative code words we predict to observe in the near future under that specific motion behaviour. The predicted qualitative code words present in $\boldsymbol{\mu}_i$ are each applied onto the metric map plane using the (x, y) coordinates of the key object and the QDC relations that hold in the code words. This results in aggregating the relations over the set of objects creating a probabilistic *target area* in the 2D map plane. In practice, this is achieved by maintaining an occupancy grid Y_i for each of the k learned cluster centres. The predicted target area is then defined as the most likely occupied region of cells in the grid under a particular motion behaviour, e.g. the target area is predicted from occupancy grid Y_i, corresponding to the i^{th} centroid $\boldsymbol{\mu}_i$. Since we have a lack of knowledge of the person's future movements, we use the occupancy grid to assign a probability of the likely outcomes onto the map plane. An example occupancy grid, Y_i, is shown in Figure 6.3, where the probability scale is shown from yellow to red. A ground truth human trajectory is shown in white, and the initial sub-sequence of poses are shown in green. This is the initial sub-sequence of poses which generated the classification into a specific cluster i, generating the target area prediction from the i^{th} occupancy grid Y_i.

In summary, the training and classification framework presented allows the mobile robot to qualitatively predict human movements within the region of space it is observing. It can make a decision on which are the most likely qualitative relations a human will achieve with respect to key objects, updating this decision given increasing number of trajectory poses. The QSR relations that represent a motion behaviour are projected onto the map plane so the robot can either approach the predicted target area to intercept the person, or move away, depending upon the setting.

6.1.3 Results and Discussion

This section contains experimental set-up and results for three experiments. However, we first present an illustration of the learned model Θ, that is, by visually inspecting the human trajectories that are assigned to three of the k learned motion patterns. These trajectories can be seen in Figure 6.4. It can be seen that the trajectories assigned to these three clusters express specific motion patterns which we associate to human behaviours or intents. For example, the trajectories in the left most image can be interpreted as movements of one specific employee who was present throughout the entire deployment, and whose desk is situated at the source of the

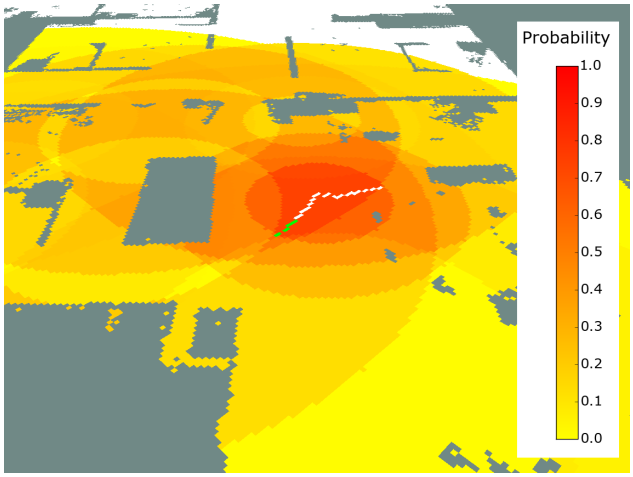


Figure 6.3: Occupancy grid and predicted target area (yellow-red) from one learned motion behaviour. A single human trajectory ground truth is overlaid in white, the first sub-sequence of trajectory poses are shown in green which are used to recognise and predict the target area. Best viewed in colour.

trajectories in this image. It shows that this employee commonly walks from behind their desk towards the door (shown in yellow), and the robot learns this as a common motion behaviour relative to the objects in the environment. The centre and right images show common motions towards (right image) and away from (centre image) the collection of desks in the right hand corner of the region. These common motion patterns can also be interpreted as behaviours, such as employees walking towards and away from their desks. These motion patterns enhance our mobile robot's knowledge of what human behaviours commonly occur in the region and each can be translated into a navigation behaviour or specific task.

Experiment A1: Classification Time

In the first experiment, we investigate real-time classification time and consistency in order to evaluate how well the system can classify a newly observed trajectory into one of the learned motion patterns using the learned model Θ. This is performed using 6-fold cross-validation where each fold is a calendar week of collected data and SC is used to learn the best number of clusters k on each fold. The classification process is performed using both QDC and QTC calculi merged together, and also when using only QDC relations. (QTC was not used alone because the QDC relations are needed to compute the predicted target area via the occupancy grid.)

For each trajectory in the test set the formulation steps are repeated to obtain an activity histogram for the first sub-sequence of trajectory poses. This is unit normalised and classified into its closest cluster centre i' and compared against the classification result i obtained when

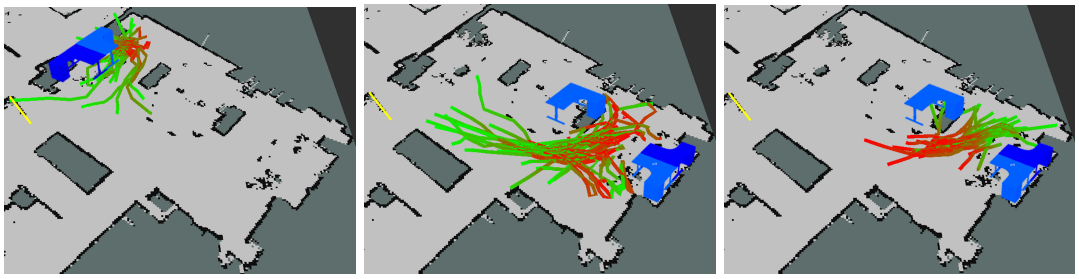


Figure 6.4: Qualitative clustering results: visualising human trajectories assigned to three learned motion behaviours, the direction of motion is shown red to green. The doorway is showed in yellow. Best viewed in colour.

the entire trajectory is used to encode the activity histogram i.e. where all future ground truth poses are used.

Results

The results of the first analysis are presented in Figure 6.5. The graphs show all metrics are marginally higher for the combination of the two calculi as opposed to using just QDC relations. It can also be seen from the graphs that once 20% of a trajectory is observed, the system has recall ≈ 0.7 (precision $\approx F1 \approx 0.7$), which demonstrates that even when only a very small section of a trajectory is observed the system is able to perform well. The metrics remain more or less at these values between $20 - 40\%$ of observed trajectory, which implies the trajectory is already somewhat abstracted and these extra poses do not improve the classification further. From 40% and above the metrics gradually increase until all the sub-sequences are observed.

To compare the two classifiers which use different calculi, QDC+QTC compared to QDC, we are interested in comparing the number of correctly classified sub-sequence instances (true positives (TP)) with the incorrectly classified instances (false positives (FP)), i.e. when the classification using a sub-sequence of poses matches the classification using the entire ground truth trajectory. There are a total of $9,074$ classifications in total across all trajectories. A 2x2 contingency table is presented in Table 6.1, where *Test 1* is the classifier based on combined calculi (QDC and QTC) and *Test 2* is the classifier based on QDC only relations.

We perform a McNemar's significance test with null hypothesis that the two classifiers have the same probability of predicting a correctly classified instance. Using a two-tailed test and a significance level (alpha) of 0.05, we achieve a Z statistic of 6.7, and therefore reject the null hypothesis. This means that the marginal proportions are significantly different from each other and validating our initial belief that QTC complements QDC very well. i.e. QDC provides qualitative knowledge about relative distances of the trajectories to objects in the region, whilst QTC provides qualitative knowledge about the relative direction of motion.

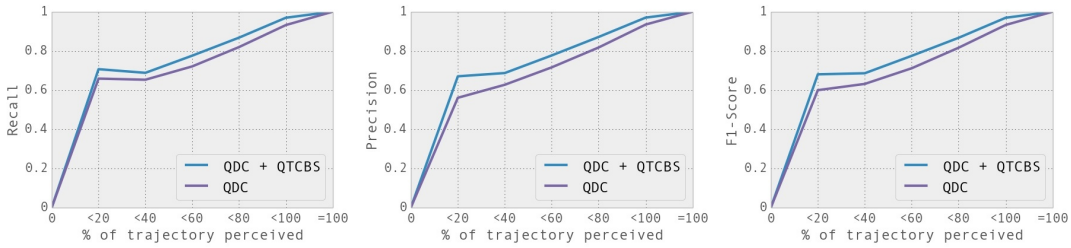


Figure 6.5: Experiment 1: Recall, precision and F1-score presented for the prediction of the correct motion behaviour of a newly observed trajectory, after obtaining a percentage of its poses.

	Test 2 TP	Test 2 FP	Total
Test 1 TP	5498	1640	7138
Test 1 FP	1278	658	1936
Total	6776	2298	9074

Table 6.1: Experiment 1: Contingency table comparing two classifiers, Test 1 uses QTC and QDC calculi combined whereas Test 2 only uses QDC. A TP is scored when a motion pattern classification given a sub-sequence of the trajectory correctly matches the classified motion pattern when the entire trajectory is observed.

Experiment A2: Predicting Target Areas

In our second experiment, we use the test set sub-sequence trajectories to evaluate how well the classification of the learned model Θ predicts a target area for future poses. Here, a previously unseen trajectory is assigned to its closest cluster centroid, e.g. $\boldsymbol{\mu}_i$ and a predicted area is calculated on the metric map using the i^{th} occupancy grid Y_i. An *occupancy likelihood score* is calculated by taking the (mean) average of the cells that correspond to the (x, y) coordinates of the entire trajectory, i.e. using the future trajectory poses. We then check if the likelihood score generated using the classified motion pattern $\boldsymbol{\mu}_i$ is greater than (or equal to) the likelihood calculated using any other learned motion pattern $\boldsymbol{\mu}_j$ for all $\boldsymbol{\mu}_j \in \Theta \backslash \{\boldsymbol{\mu}_i\}$.

Results

The results of our second experiment are presented in Table 6.2. Note that the optimal value for k, calculated using SC, changes between the different training sets and is presented in the table. The average scores of recall 0.53 (precision 0.67 and $F1$ of 0.55) demonstrate that, for more than half of new trajectories, the predicted target area generates the highest likelihood score when compared to the other learned motion behaviours. Given the challenging nature of the data (as will be proved later) these results show good performance. This implies that our unsupervised qualitative learning k-means framework is able to express the human motion patterns that emerge in the environment and predict reasonably well the expected future lo-

cations of trajectories using the generated occupancy grids. Results of the same classification task using Random and ZeroR classifiers are also presented as a baseline.

cv-fold	k	recall	precision	$F1$
week 0 (342)	13	0.48	0.59	0.48
week 1 (169)	9	0.56	0.80	0.62
week 2 (196)	12	0.48	0.65	0.51
week 3 (104)	10	0.63	0.75	0.67
week 4 (205)	11	0.53	0.53	0.52
week 5 (216)	11	0.48	0.67	0.51
Avg:	11	0.53	0.67	0.55
Random:	–	0.08	0.10	0.08
ZeroR:	–	0.21	0.04	0.07

Table 6.2: Experiment 2: Testing the maximum occupancy likelihood score (of all k motion behaviours in Θ), against the classified motion for trajectories in the test set.

Furthermore, the average recall of the system increases to 70% (precision 0.80 and $F1 = 0.72$) if we consider the highest two occupancy likelihood scores. This is particularly relevant because the predicted target area, given by the occupancy grid Y_i, is computed using QDC relations only. However two occupancy grids could appear similar, when their underlying motion behaviour differs due to different QTC relations. For example the two motion behaviours, centre and right, in Figure 6.4 are overlapping with respect to qualitative distance relations to key objects (QDC) only, but are qualitatively different when considering the direction of motion relations (QTC) as they have opposite directions.

Experiment A3: Increasing Trajectory Data

Finally, we investigate the effect of adding additional training data into the system. This aims to mimic a live deployed robotic system as it accumulates training data over a long term deployment, i.e. the ability to predict target areas for new trajectories should improve with more training data. Here, we use the six weeks of data and use the final week as a constant test set to evaluate. We repeat Experiment A2 using the first five weeks of trajectories incrementally added into the training set, repeating the learning process five times in total.

Results

Table 6.3 presents the results of the prediction task, replicating a live deployment setting. We repeat Experiment A2 with accumulated training data over the first five weeks testing on the sixth, allowing k to vary for each training phase separately. The results show that as more training data is acquired and more human trajectories are represented qualitatively, the system is better able to classify new trajectories into its learned motion behaviours. This is further emphasised considering Week 0 contained more trajectories than any of the other five weeks.

Training Weeks (M)	k	recall	precision	$F1$
week 0 (342)	9	0.24	0.72	0.29
weeks 0-1 (511)	12	0.43	0.54	0.44
weeks 0-2 (707)	12	0.43	0.56	0.43
weeks 0-3 (811)	10	0.43	0.71	0.49
weeks 0-4 (1016)	14	0.48	0.63	0.53
Avg:	11	0.40	0.63	0.44
Random:	–	0.08	0.10	0.08

Table 6.3: Experiment 3: Maximum occupancy likelihood score (testing all k motion models Θ) matching the classified motion, using cumulative training data.

Literature Comparison

We aim to validate the results presented in this section obtained from our qualitative framework for learning motion behaviours. A comparison to the popular quantitative approach presented in the literature [Bennewitz et al., 2002] is proposed using their *Expectation Maximisation* (EM) framework to predict human motion patterns from mobile robot trajectory observations. The goals of their work overlap with our main objectives. They present an algorithm that "learns collections of typical trajectories that characterize a person's motion patterns." It fits a Gaussian Mixture Model (GMMs) over the exact (x, y) locations of human trajectories of equal length.

One practical difference between the trajectory observations in their work is that they use a multi-robot set-up to obtain near-complete trajectories from a source to a sink location. This results in many trajectories, all of (qualitatively) similar length through the environment. To replicate this, our training dataset of noisy, partially observed trajectories are extrapolated to the maximum trajectory length in order to fit the GMM. This was performed, for trajectories with length less that the maximum, by repeating the final trajectory pose; simulating the person was detected stood still at the sink location. In our dataset, the maximum number of poses is 420, equivalent to roughly 16 seconds of observation in the robot's field of view.

The published EM algorithm was unable to successfully converge when using our trajectory observations, possibly due to the noisy nature of the data, or the extrapolation. We repeated the experiment using a subset of the training dataset of $1,232$ instances from the experimental region only. The EM algorithm failed to converge in reasonable time when initialised with a sensible starting number of Gaussian Mixtures. The iterative procedure continued to add a *motion pattern* to the model due to low data likelihood, then remove one, due to low *motion pattern utility* which tests each motion in turn and indicates where a similar motion pattern exists and they are duplicates. This comparison with a popular technique in the literature highlights the difficulties in scalability of quantitative approaches and the challenges of clustering real-world trajectory data in an unsupervised setting and learning coherent motion behaviours.

Summary

Our training and classification framework presented in this section allows a mobile robot to learn a set of motion patterns from incomplete and partial trajectory observations, and use them to predict future human movements. It characterises human motion behaviours and is sensitive to key objects in the region, whilst being invariant to exact metric positions. This is a challenging task given a real-world deployed mobile robot is not able to perceive complete human trajectories from their source locations and is compounded when the robot is required to make real-time predictions about a human's future movements using only the initially observed trajectory poses.

We showed that QDC and QTC calculi complement each other well for the task of learning human motion behaviours in an indoor environment, and provide different modalities of qualitative information about the motions. We demonstrated that the robot is capable of predicting the likely area to be occupied of a newly observed trajectory. That is, the robot can make a prediction on what are the most likely qualitative relations a human will achieve with respect to key reference objects, continually updating this decision whilst more trajectory poses become available. These relations are projected onto the 2D map plane so the robot can either approach the target area to intercept the person, or move away, depending upon the setting. Finally, it is shown that the performance increases as it accumulates more training observations, but is restricted to running in a batch manner, i.e. recomputing the cluster centroids each time.

In the next section we introduce a more detailed human body pose dataset. This requires the use of more sophisticated learning methods in order to learn more detailed human activity classes equivalent to distributions over the discrete vocabulary, and where each observation is represented as a mixture over the classes.

6.2 Learning Activities from 3D Body Pose Sequences

Given more detailed human body pose observations, it is possible for the robot to learn common patterns and behaviours corresponding not only to human motion on the 2D map plane, but to patterns of body joint movements in 3D space with respect to key objects in the environment. This allows the robot to learn human activities at a more fine granularity to those restricted to the map plane. Understanding human activity at different granularities can aid the robot in understanding how space is used and can drive particular robot behaviour such as exploration or human-robot interaction. A likely future goal could be for the robot to share or help the human towards a common goal in a newly learned activity.

In this section, we introduce two human pose datasets consisting of RGBD images and human body pose estimates: 1) The first dataset we use is a publicly available and popular dataset from the literature consisting of 10 scripted daily living-type activity classes recorded from a static camera location. Each activity class is repeated ≈ 12 times by four volunteers and annotated with a corresponding ground truth class label. Many videos contain real-time

object detections at each timepoint and correspond to dynamic objects in the scene which allows us to test our qualitative framework using dynamic object locations as well as static ones. However, this dataset is performed by volunteer actors where a single staged activity is performed repeatedly in front of a static camera with fixed view point to obtain the clearest possible human pose estimates. 2) Secondly, we present a dataset observed and recorded from a mobile robot observing an unstructured, real-world university common room and kitchen area over a one week duration. In contrast to the first dataset, this is not scripted in advance and recorded humans perform activities in real life situations. It contains a variety of human activities observed from multiple viewpoints, for example, heating food, preparing hot drinks, using a multi-function printer, throwing trash and washing up. This dataset also consists of the map frame locations of key objects in the environment learned by performing 3D sweeps and segmenting out object clusters.

First we present the details of each dataset and then describe the implementation details used to compute activity histograms and a term-frequency matrix. We show that sophisticated unsupervised learning methods, such as LSA and LDA can learn consistent activity classes from both datasets when compared to the human-annotated ground truth labels, and that the performance of LSA and LDA is superior to simple clustering approaches in this more high dimensional and complex setting.

6.2.1 Cornell Activities for Daily Living Dataset

We introduce a popular and freely available human activities dataset consisting of activities for daily living from Cornell University as a benchmark to evaluate our unsupervised learning framework [Sung et al., 2014]. This dataset consists of 124 RGBD videos, acted out by four actors, where each performs one of 10 high level activity classes in each video clip, resulting in 124 short video sequences. The activity classes are predefined as: *arranging objects*, *cleaning objects*, *having a meal*, *making cereal*, *microwaving food*, *picking objects*, *stacking objects*, *taking food*, *taking medicine* and *unstacking objects*.

The dataset has a hierarchical structure and each of the high level activity classes is comprised of multiple lower level activities, such as: *pouring*, *eating*, *opening*, *placing*, *reaching*, *moving*, *cleaning*, *drinking* and *closing*. The occurrence of these overlap within the 10 high level activities, which provides some intra-class similarities. The activities are performed facing a fixed camera, with little or no background clutter and with the subject in the centre of the frame. Each high level activity is repeated three times per actor (four times for *making cereal*), creating 124 high level activity instances.

Example images from the dataset are shown in Figure 6.6, where the bottom row images show an example of the "making cereal" activity class where the minimum bounding rectangles (MBR) for the tracked objects are overlaid along with a sub-set of the human body pose estimate (as red points and green lines) onto the RGB image. The Cornell Activity Dataset provides skeleton pose estimates consisting of 15 joints positions along with auto and ground

truth object detections and tracks. A key challenge in this dataset is that certain key objects in the environment are dynamic and tracked during the observation resulting in a sequence of (x, y, z) object poses (one per timepoint). The objects are not always at the same exact location across multiple videos, but each object detected is semantically tagged with an object class, such as, "microwave" or "cereal box" which allows us to encode the abstract object type into our qualitative representation.

This publicly available dataset is considered one of the largest and more challenging vision datasets of humans activities in recent literature. It contains real-world activities that occur in one's daily life and are considered useful for a robot to learn about. Further, the human body poses are estimated using OpenNi which struggles with body joint occlusions, even with reaching or placing activities. However, it is not considered complex compared against a robot's 'in-the-wild' human observations. The activity classes are clearly defined, scripted and acted with slow body joint movements. It has very balanced activity classes in terms of the number of instances within each classes; ≈ 12 repeats of each. Also, the videos are perfectly temporally localised focusing only on a single activity instance. Finally, there is high inter-class similarity,

Figure 6.6: Example images taken video clips in the Cornell Activities for Daily Living (CAD120) Dataset [Sung et al., 2014]. Bottom row represents three images of "making cereal" activity class where the human pose estimate and object detections are overlaid. Best viewed in colour.

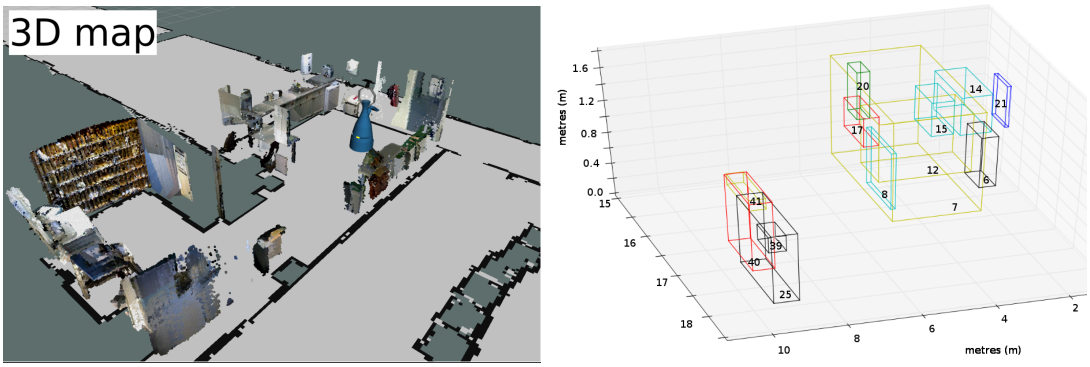


Figure 6.7: Human Activity Dataset environmental set-up and autonomously learned object locations. (left:) 3D object clusters extracted from fused point clouds, overlaid onto the metric map. (right:) sub-set of object clusters selected using analysis of trajectories. The axes refer to the metric map frame relative to the map origin.

i.e. the repeated activities are performed by 4 participants usually in a very similar visual manner; although one participant is left handed.

The rationale for including analysis on this fixed camera staged dataset, is that it provides a known set of activity classes at a particular activity granularity in order to test the qualitative framework presented in the thesis. Further, the availability of dynamic object tracks also provides an interesting extension to using static objects learned from a mobile robot.

6.2.2 Leeds Human Body Pose Dataset

Here we introduce a real-world human observation dataset captured by a Metralabs Scitos A5 mobile robot deployed in a natural human-populated environment. The human body pose dataset is captured by observing members of staff and students performing every day activities in the kitchen and student common areas mapped by the robot. No restrictions are placed upon the observations or activities that occur in the environment, meaning that the robot observes many partial, incomplete, and fast-paced interactions between people and objects, where body poses are challenging to estimate. The dataset also provides many difficult variations due to lighting, various viewpoints and many occlusions, usually from multiple people in the environment at the same time. The dataset collected, along with meta-data and software repository, is available at: http://doi.org/10.5518/86.

Objects: The robot first performs a 3D metric sweep of a university kitchen and student room in order to generate a registered 3D point-cloud of the target environment as described in Section 3.2.3. The robot extracts a set of key object clusters from this representation and overlays them onto the metric maps. The object clusters learned using this process can be seen as registered point cloud points in Figure 6.7 (left). A sub-set of objects are selected using

the analysis of human trajectories, i.e. where people stop and what locations their body joints interact with. This resulted in a set of 41 key object locations shown in Figure 6.7 (right), that somewhat correspond to locations of real-world objects, e.g. object ID 6 corresponds to the trash bin, 21 to the paper towel dispenser, and 25 to part of the printer-copier machine. The locations of the autonomously learned key objects are evaluated with respect to manually specified objects that people interact with in [Alomari et al., 2017a]. This evaluation provides a set of 12 manually specified key object locations encoded as SOMa objects and which we can also use to compare our learning framework to determine how dependent it is on accurate object positions. The manual objects include: *shelves, microwave, water cooler, tea/coffee pot, sink, kettle, fridge, paper tray, printer screen, paper towel dispenser* and two *waste bins* and can be seen in Figure 3.5.

Human Body Poses: The robot was tasked with patrolling pre-defined topological nodes and observing the environment with various different viewpoints. Given a detected person in the robot's field of view, the camera records RGB images along with the estimated human body pose as described in Section 3.3.2. The human body pose is estimated from the camera image, first using the depth image (OpenNi2), then the RGB image post-process (CPM). Each body joint position is then translated into the robot's map coordinate frame of reference using the localised position of the robot and the pan-tilt angle, i.e. where the camera is pointed. Obtaining an accurate position of the body joints in the map frame relies on the robot being well localised within the map. The visual SLAM algorithm is not always accurate when the robot is moving, so we restrict the human observations to when the robot is static.

The dataset was collected over the period of one week. The robot's schedule randomly selects between the set of topological waypoints to visit and observes the environment. The robot observed 287 individuals during the process and estimated a human body pose sequence for each. These sequences contain arbitrary number of poses with high variance, with an average (mean) number of 513 poses and standard deviation of 588 poses, indicating a very large spread, representative of the nature of the observations. A selection of example detections can be seen in Figure 6.8.

For the purpose of obtaining ground truth (GT) activity class labels for the human observations, each recorded sequence was manually inspected by volunteers. A set of common and repeated activities present in the recordings was agreed upon and this defined the set of activity classes annotated. The activity granularity of the defined classes was somewhat limited by the data available from the robotic vision component; in particular no object tracker was available, and hand tracks are not always reliable, this means that activities involving small objects or complex hand movements were undistinguishable by the visual component, and thus only human activities involving static objects were used as ground truth labels.

The occurrences of each instance of each activity class within the observed videos was temporally segmented by the volunteers, creating multiple shorter video sequences containing

Figure 6.8: Example RGB images with human pose estimate overlaid obtained from the Leeds Human Activity dataset recorded from a mobile robot. The bottom row shows instances of the same activity class observed from multiple viewpoints. Best viewed in colour.

only a single activity class in each. Table 6.4 presents the list of the activity classes annotated, along with the number instances in each class. A total of 493 individual activity class instances were extracted, with 77 observations containing no activity classes. These sequences are much shorter, with average (mean) number of 137 poses and standard deviation of 191 poses, and temporally focused on the activity instance taking place. We consider these sequences as each containing a single *interesting* human activity, as defined and segmented by the volunteer annotators.

It is these segmented clips which form the basis for the experiments in this section. As anticipated, the dataset is highly unbalanced with respect to the number of instances of each class observed, i.e. some activities were observed more frequently than others pertaining to the fact these activities occur more often in the environment. Further, the durations of each instance within a class can vary greatly also. That is, the dataset provides large inter and intra-class variation. This is somewhat highlighted in the sample images in the bottom row in Table 6.4; multiple different viewpoints of the same activity class are often detected. Here the observed person is "using the printer interface", recorded from three different locations. Another key challenge when using this dataset is that many of the activity classes occur within

Activity Class	# Instances
Wash cup	82
Take object from fridge	81
Use the kettle	70
Throw trash in bin	65
Take paper towel	45
Take tea/coffee	35
Use printer interface	35
Use the water cooler	26
Take printout from tray	24
Microwave food	19
Opening double doors	11
No Activity Label (N/A)	77
Total	493 (570)

Table 6.4: Leeds Activity Dataset: Annotated activity class labels, with corresponding number of instances segmented. "N/A" is a human observation with no activity occurring defined by volunteer annotators. The "Total" is shown excluding and including (in brackets) the N/A labelled videos.

a relatively small-spatial radius within the environment, meaning the activities are challenging to temporally segment since they can often overlap within the observation. This is a major challenge using real-world recorded human observations and where the activity granularity of observations is not pre-defined.

Finally, note that 77 (out of all 287) observations/recording were deemed to contain none of the above activity classes by the annotators, and given the label "NA". This is a considerable percentage of noisy observations and provides another major difficulty when no manual segmentation or filtering of the data is provided. We propose an approach for handling this kind of data obtained 'in-the-wild' in the next chapter.

Human Body Pose Filtering: To reduce the effect of both robot localisation errors and body pose estimate errors in the camera frame, we implement a median filter applied to the location of the estimated human body joints across the sequence of poses recorded [Jones et al., 2001]. This helps to somewhat smooth the change in location of each body joint and has the effect that a detected body joint cannot suddenly "jump" far away from its previous location in space. The hypothesis is that if a body joint location moves too far over a small window of frames it is a result of a robot localisation or pose estimation error. We used a window of 10 poses/frames in this case.

As an example of the filtering process, Figure 6.9 shows the x and y camera frame position of a human body joint (the right hand) in a sequence of 200 detected body poses. That is, the x (left) and y (right) camera frame coordinates of the estimated right hand location are plotted over the sequence of poses from 0 to 200. The CPM estimates the body pose per detected image separately and therefore does not attempt to smoothly track body joints across

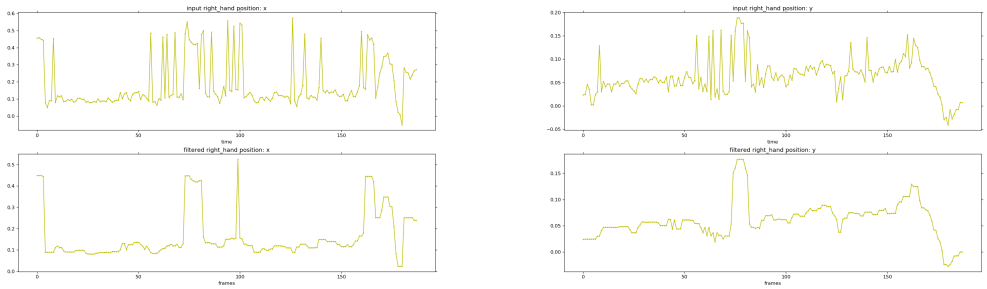


Figure 6.9: Example estimate human body joint pose filtered using a median filter with window of 10 poses. (top:) original camera frame position of right hand body joint. (lower:) filtered position of right hand body joint. Each axis is filtered separately: x (left), y (right).

consecutive frames like more sophisticated approaches. For this reason it can be seen that a joint pose estimate is not always smooth across frames, i.e. an estimated body joint location, such as the hand in this plot, can be detected to oscillate very quickly, which is unlikely to represent the true pose of the body joint; detections such as these are assumed a result of a pose estimate error. In order to smooth these estimates a median filter with window size of 10 poses is applied to each axis separately, and can be seen in the bottom row of Figure 6.9. We can see that the filter removes some oscillating behaviour from the estimation, and smooths the directional changes of the body joint poses. The window size was selected experimentally in order to capture the actual body joint movements.

6.2.3 Implementation Details

Here we present the implementation details for applying our qualitative relational framework to the two human body pose datasets in order to learn coherent human activity classes in an unsupervised setting. We first present the common implementation details applied to both datasets, such as, the qualitative representations and code word parameters which are common, ultimately, leading to activity histograms computed for each observation and a two term-frequency matrices (one per dataset). We then describe the specific details of each term-frequency matrix, and finally discuss the specific parameters for the different unsupervised learning methods implemented. In the next section we describe the experiment set-up and present the results for both datasets.

For an observed and segmented video clip m in the dataset, it is first represented as a human pose sequence $S_m = [p_1, \ldots, p_i, \ldots p_t]$ of length t, where each p_i is the human body pose at timepoint i and contains both the camera frame and map coordinate frame 3D position of 15 body joint locations, i.e. as 15 joint poses. Recall that no restrictions are placed upon t, it is arbitrary and varies for each observation. Abstracting this sequence of body poses into a qualitative representation is performed in a two stage process. First, the TPCC calculi is used

to abstract the person's relative body joint positions relative to the head-torso (origin-relatum) 2D line in the camera coordinate frame, i.e. in the image plane. Secondly, the QTC and QDC calculi are used to abstract the person relative to the key objects in the map coordinate frame. The details of both processes are given next.

Camera Frame QSR: Qualitative TPCC relations are calculated for each pose in the body pose sequence by fixing the *origin* and *relatum* to the *head* and *torso* joint positions respectively to generate a person's "centre line" at each pose. The sequence of TPCC relations Q_{cam} of length t contains TPCC relations between the centre-line and the left/right hands and shoulders joint positions. Other joints were omitted for efficiency and also because we assume their movements do not contribute to the kinds of human activities in our two datasets. However, they could obviously be added. This relates to 4 rows in an interval representation where each row corresponds to TPCC relations holding between the centre-line and one of the selected body joints.

Map Frame QSRs: To abstract the body joint locations in the map coordinate frame we use a combined QDC and QTC calculi. Similar to the abstraction of human trajectories, QTC captures relative motion and QDC captures relative distances of the poses to key objects; however, unlike the trajectories, the poses relate to individual body joint movements not just 2D map plane detections. These QSRs are used to represent the relative movements of the person's left/right hand body joints and torso location, relative to the key objects. A sequence of combined QDC and QTC pairs is produced, Q_{map}, of length $t-1$ since QTC relies on pairs of consecutive poses so the QDC value at $t=1$ is removed to obtain $t-1$ QSR pairs. QDC thresholds are used and the specific values are given below for each dataset. However, by using semantically meaningful QDC relations in the sequence Q_{map}, we do not encode intervals for any timepoints where the QDC value is greater than the largest boundary; this is considered as a QDC value of "ignore". This has the effect of producing a sparse interval representation leading to a more efficient process and ultimately, fewer discrete code words encoded per observation. The QTC "quantisation factor" which relates to the minimum change in pairwise distance between consecutive timepoints to accept a relational state change is set to 0.1 and a conceptual neighbourhood is not used to insert unobserved relations (as this would add new timepoints into the QTC sequence, but no corresponding QDC relation to form QTC-QDC pairs). For each sequence Q_{cam} and Q_{map}, we apply a median filter (window size of 10), which smooths rapid flipping between relations. This is not uncommon when abstracting into a qualitative representation, for example, pair-wise distances that exist on or close to a QDC boundary can constantly flip between relations. The aim is to retain only those relational changes which form intentional human body movements.

We create an interval representation and interval graph for each sequence separately and extract all graph paths from both using graph path parameters ρ and η given per dataset below.

Since the qualitative relations differ between the two sequences, we can merge the two collections of extracted graph paths together in order to create a single bag-of-qualitative-words to represent the observation, i.e. $d = \{w^1, w^2, \dots\}$, where each w^i is an observed discrete descriptor (graph path) extracted from one of the two encoded interval graphs, and not necessarily unique.

Cornell Activity Dataset: In each video sequence in the CAD120 dataset, the person is situated near the centre of the camera frame, performing the activity and the objects were situated close to the person. For this reason, the QDC relation thresholds used in this case are $\Delta = [0.15m, 0.4m, 0.8, 1.0m]$ creating five semantic regions which can be labelled as *touch* [0-0.15m], *near* (0.15-0.4m], *medium* (0.4-8.0m], *far* (0.8-1.0m] and *ignore* (>1m). These were experimentally chosen to distinguish the body pose movements in the simple set-up. Also note, for the dynamic objects, the abstract object class is used as the object ID in the interval representation and interval graph.

For each of the 124 training videos in the CAD120 dataset, a bag-of-qualitative-words, d, is computed and the union of all these bags forms a code book vocabulary $\mathcal{V}_D$ of unique code words (graph paths). In this case, $|\mathcal{V}_D| = 5,520$ unique code words using the graph path parameter choices: $\rho = 3$ and $\eta = 1$. Finally, an activity histogram is computed for each video and an $M \times N$ term-frequency matrix is computed where $M = 124$ and $N = 5520$, with a total of $29,016$ code words observed in total.

Leeds Activity Dataset: The video sequences recorded from the mobile robot are much more varied and challenging. We evaluate the learning methods when using two sets of keys objects. First, the set of 14 most interacted with key objects are obtained from the 3D sweeps and trajectory analysis; these align reasonable well to real objects in the environment, and they are not labelled with any prior semantic knowledge. Secondly, we evaluate using the set of 12 manually specified key object locations as encoded as SOMa objects and described in Section 6.2.2. This allows us to determine how important obtaining the exact location of key objects is to our framework. QDC thresholds used in this case are $\Delta = [0.25m, 0.5m, 1.0m]$ creating four semantic regions which can be labelled as *touch* [0-0.25m], *near* (0.25-0.5m], *medium* (0.5-1.0m] and *ignore* (>1m). In our case, these were experimentally chosen to distinguish activities in this more complex environment, however, it is possible to learn the threshold values from observations in an unsupervised setting [Behera et al., 2012a].

Similarly to the CAD120 dataset, a bag-of-qualitative-words is computed for each of the 493 segmented video clips and a code book vocabulary $\mathcal{V}_D$ is generated from the unique code words using maximum path-length $\rho = 4$ and restricting the nodes on a path to encode at most 2 object pairs, i.e. $\eta = 2$. Here, using the 14 autonomously learned key objects, $|\mathcal{V}_D| = 20,637$. A low-pass filter removes code words from the vocabulary if they are not observed in at least 5 observations, reducing this very large vocabulary to a more manageable $N = 2,876$. Similarly, $|\mathcal{V}_D| = 22,829$, reduced to $3,594$, for the case when using the set of manually defined SOMa key

objects. An activity histogram is computed for each video (and each set of objects) resulting in the $M \times N$ term-frequency matrix. We present results using the reduced term-frequency matrices for all the learning methods in the next section.

6.2.4 Results and Discussion

In this section, we present experiments and empirical results to validate the qualitative, unsupervised learning methodology for challenging human observations presented in this thesis. We demonstrate this by applying our learning framework and experimental procedure to the two human activity datasets described above. The structure of this section is as follows: first we introduce the cluster metrics which we use to evaluate how coherent the learned classes of human activity are compared to the ground truth labels, then we present the results of the different unsupervised learning methods on both datasets. Our proposed unsupervised learning methods are supplemented by a comparison to a commonly used supervised learning technique, a Support Vector Machine (SVM) and the simple unsupervised clustering technique k-means. Lastly, we discuss how we can interpret the learned topics and how they represent human activities described over the discrete vocabulary.

Cluster Metrics

Unsupervised learning methods do not use the ground truth label assignments of any of the training samples like supervised approaches do. This means it can be challenging, and not always suitable, to map the learned classes directly to each ground truth class. In our case, after the training phase, a sample can be represented as its closest cluster centroid (k-means), a linear combination of latent concepts (LSA) or as a multinomial distribution over topics (LDA), and so a many-to-many mapping can exist between the set of ground truth labels and the emergent classes. This problem is especially pertinent when dealing with highly unbalanced classes in the training data, which is indeed the case for the Leeds Activity dataset. Therefore, we provide results using unsupervised clustering metrics where the aim is to test how coherent the emergent classes are with respect to the ground truth labels, i.e. is a cluster composed of data points all with the same ground truth label, or that all instances of the ground truth class are present in the same emergent cluster. For this purpose we use two metrics, V-measure [Rosenberg and Hirschberg, 2007] and (Normalised) Mutual Information (NMI) [Vinh et al., 2009]). Both metrics provide a score of how closely two sets of labels match for the same set of data. We use these to compare the ground truth labels (assigned by volunteers to each observation), to the emergent class assignments from the learning process. If the assignment is multinomial, we select the class with the highest probability for this experiment.

The V-Measure is a combination of the *homogeneity* and *completeness* clustering metrics, given two sets of labels. Homogeneity evaluates whether the predicted clusters contain only data points which are members of the same ground truth class; whereas completeness evaluates

whether the member data points of a given ground truth class are all elements of the same predicted cluster. Both values range from 0 to 1, with higher values desirable. The V-measure is computed using: $v\text{-}measure = 2[(homogeneity \times completeness)/(homogeneity + completeness)]$.

The second popular metric for unsupervised learning is the Mutual Information (MI) score. It can be computed with the following formula:

$$MI(U,V) = \sum_{i=1}^{N}\sum_{j=1}^{N} P(i,j) log \frac{P(i,j)}{P(i)P'(j)},$$

where, U and V are two lists of N labels, $P(i)$ is the probability that a random sample occurring in cluster U_i and is calculated using: $P(i) = |U_i|/N$. Similarly, $P'(j)$ is the probability of a random sample occurring in cluster V_j and $P(i,j)$ is the probability that a sample picked at random falls into both classes U_i and V_j, i.e. the intersection $|U_i \cap V_j|/N$. The Normalised Mutual Information (NMI) is computed by normalising the MI by the entropy (the amount of uncertainty) for each of the partitions in each list of labels. The equation is given in Appendix C.6. NMI can be thought of as a measure of how many bits are needed in order to store predicted outcomes given that the true value is known and it provides a measure of similarity of any two sets of class labels, where 0 indicates no mutual information and 1 indicates perfect correlation.

Experiment B1: Learning Body Pose Activity Classes

Given an $M \times N$ term-frequency matrix C representing one of the training datasets, we learn activity classes and compare them to the ground truth annotated activity labels. We use the three learning methods introduced in Chapter 3.

LSA: For Latent Semantic Analysis, first we apply the binary low-pass filter over the term-frequency matrix to remove any code word that occurs in fewer than 5 observed activities. Then the tf-idf weights are computed and applied to the term-document matrix in order to perform the SVD decomposition on these new values weighted be their occurrence across the entire corpus.

The SVD decomposition recovers the singular values in the diagonal eigenvalues matrix. An example of the singular values extracted from the Cornell daily living dataset is plotted in Figure 6.10, showing 124 singular values. Recall the rank of the term-frequency matrix C is equal to at most the minimum dimension, $min(M,N)$, i.e. 124. This decomposition is used to highlight a suitable rank, r in order to compute a low-rank approximation to our original term-frequency matrix C, and therefore the appropriate number of emergent concepts to learn. In this example it can be seen that there are a small number of "large" singular values where each could represents a latent concept present in the encoded matrix; this relatively small number is expected given this dataset contains only 10 ground truth activity classes, although there

could be multiple ways of performing the same activity. We threshold and use only the largest singular values (the best threshold is shown as a green vertical line in Figure 6.10) and present the cluster results for the first and most significant 10 latent concepts in the next section. For the Leeds Activity dataset, 11 large singular values was found to be the optimal using this method.

LDA: For Latent Dirichlet Allocation, we set the number of topic distributions to the values chosen when using LSA, 10 and 11 for the CAD120 and Leeds Activity datasets respectively. However, we propose a method to alter this number dynamically during an incremental learning process in the next chapter. Recall that for each observation the generative LDA model samples a topic proportions vector, $\boldsymbol{\theta}$, from a prior Dirichlet distribution, i.e. $\boldsymbol{\theta} \sim Dir(\alpha)$ and similarly, for each topic, samples a multinomial distribution over the code book $\boldsymbol{\phi}$ from a Dirichlet parametrised by β. Therefore to motivate the choices of α and β , we demonstrate three samples drawn from a 10-dimensional Dirichlet distribution parametrised by the hyperparameter π, which is arbitrarily set to six different values: 0.001, 0.01, 0.1, 1.0, 10 and 100 in order to understand the samples drawn. The drawn multinomial samples are shown in Figure 6.11 where the value of π increases from left to right.

The multinomial samples clearly show that when π increases, the mixture proportions vector $\boldsymbol{\theta}$ is more uniform over the topics. This is important, since it is draws from the multinomial distribution $\boldsymbol{\theta}$ which define the topic assignment $\mathbf{z}$ tokens that specify which topics to sample in order to obtain code words. Increasing the hyperparameter π therefore means that topic assignments drawn from $\boldsymbol{\theta}$ are more likely to correspond to multiple topics and not all from a single topic. Likewise, if π is set very small, all assignments $\mathbf{z}$ would represent a single topic and the observation would be assumed to be generated from code words sampled from only

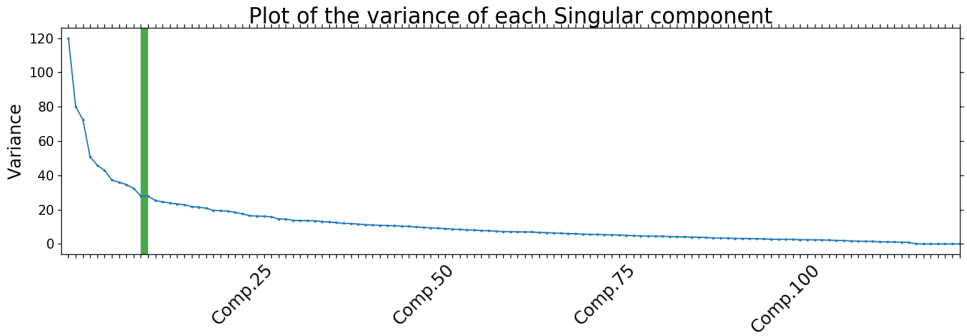


Figure 6.10: Variance of singular values of the LSA decomposition on the CAD120 dataset, encoded as a term-document matrix C in Experiment B1. The x-axis represents the singular values (components) ordered by their variance, to the maximum of rank(C). The chosen threshold limit is shown as green vertical line.

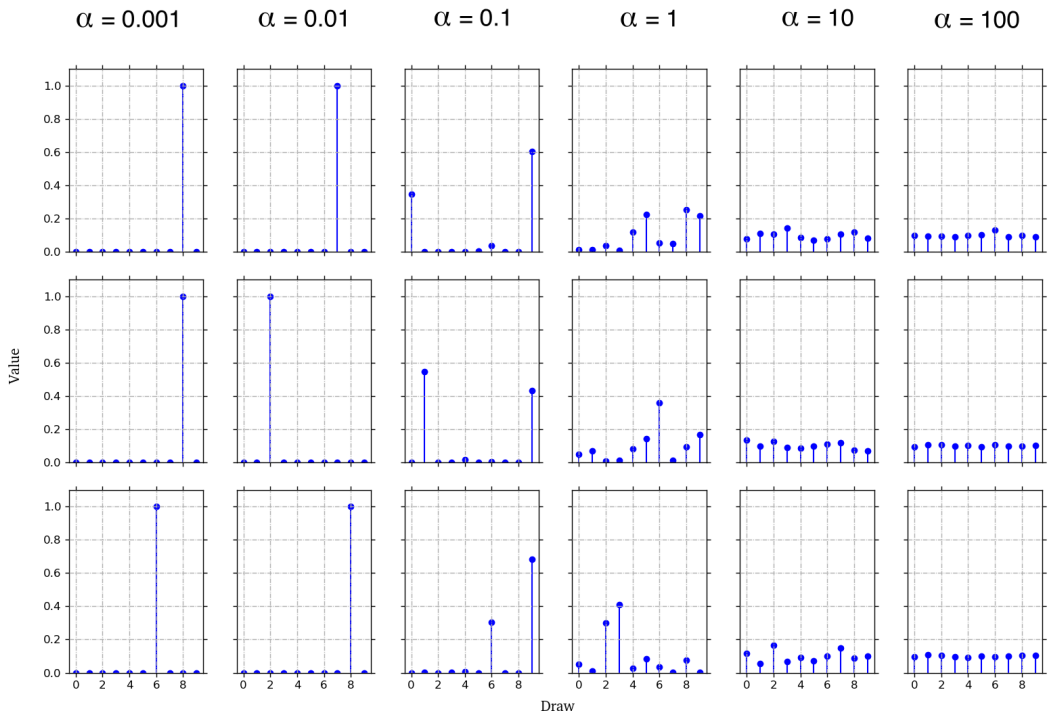


Figure 6.11: Example samples from a 10 dimensional symmetric Dirichlet distribution, drawn from $Dir(\alpha)$ with hyperparameter $\pi = 0.001$, 0.01, 0.1, 1.0, 10 and 100. As π increases, the simplex becomes more peaked and the multinomial distribution samples become more uniformly distributed across the 10 dimensions.

a single topic. Although the videos in both datasets are temporally segmented, with a single activity instance in each, we set the hyperparameters α, β to 0.5 and 0.03 respectively. This means the $\boldsymbol{\theta}$ distributions will allow for mixing of a small number of topics in each observation, which is ideal since some activity classes are quite similar and have overlapping qualitative code words. Modelling each observation in this way allows for the emergent topic distributions to correspond to activities even when the observation contains code words highly probably across multiple classes. Similarly, the code words sampled from the N-dimensional topic distribution $\boldsymbol{\phi}$ will be mixed since it is more suitable to represent each human activity class over a number of discrete qualitative code words.

Upper and Lower Bounds: To add perspective to the human activity classes learned using our proposed unsupervised framework, we present multiple comparison techniques. First, we compare against the simple unsupervised clustering k-means algorithm, where the number of clusters is set to the optimum number found for LSA. Due to variations in the initialisation of the algorithm, the presented results are an average over 10 repeated runs, each initialised randomly.

Metric	Random clustering	Unsupervised k-means	Unsupervised LSA	Unsupervised LDA	*Supervised* SVM
V-measure	0.18	0.63	0.76	0.74	0.84
Homogeneity Score	0.18	0.69	0.75	0.72	0.84
Completeness Score	0.18	0.58	0.77	0.77	0.84
Mutual Information	0.41	1.33	1.72	1.66	1.93
Normalised MI	0.18	0.63	0.76	0.75	0.84
Accuracy	0.10	N/A	N/A	N/A	0.81

Table 6.5: Experiment B1 results for CAD120 dataset: Cluster metrics obtained comparing the ground truth labels of 124 segmented video clips encoded using a qualitative framework, against the learned, emergent human activity classes. The table shows methods of increasing sophistication: unsupervised k-means clustering; low-rank approximate LSA; Generative LDA; compared against random chance and a supervised SVM as an intuitive lower and upper bound respectively.

Secondly, we propose a supervised learning technique as a hypothetical upper bound on performance. We learn human activity classes using a supervised Support Vector Machine (SVM) algorithm on the rows of the encoded training term-frequency matrix with corresponding ground truth labels. The SVM is trained using 5-fold cross validation, with a linear kernel, and where the code book is trained once across the whole dataset (as opposed to recomputing the code book at each training fold, with the effect that a code word may not be observed in the training but only in the test data.). The SVM fits high-dimensional decision boundaries between the labelled training samples. This supervised technique has access to the ground truth labels during the learning process and so we naturally expect its to out-perform the unsupervised approaches. Finally, we present the results of random clustering as an average over 10 repeated runs as a lower bound. We expect each of our proposed learning methods to perform better than this.

Results: Cornell Activity Dataset

Results for Experiment B1 applied to the Cornell Activities for daily living dataset are presented in Table 6.5. We present the cluster metrics for each of our three proposed unsupervised learning methods, a random clustering baseline and a supervised SVM approach considered as a upper limit in performance. For LSA specifically, after the low-pass filter is applied, the 124×5520 term-frequency matrix C is reduced in size to 124×958. Note that applying a low-pass filter improves the results for the LSA, but for relatively small number of code words, it does not improve the LDA therefore we present LDA results on the full term-frequency matrix. For the case where a video is assigned a multinomial distribution, the highest value is taken as its classified topic. The results clearly show that the more sophisticated learning methods, such as LSA and LDA, are able to learn coherent clusters that correspond well to the ground truth activity classes. Similarly, the supervised SVM performs slightly better, however it uses the ground truth labels to compute its decision boundaries.

All 124 videos have been classified and the unsupervised LDA results are further presented using a confusion matrix of the emergent topic assignments vs the ground truth class labels and is shown in Figure 6.12. The confusion matrix shows that when using the unsupervised LDA, some activities such as *making cereal*, *taking medicine*, *microwaving food*, *taking food* and *cleaning objects* are separated very well and all instances in the dataset are correctly classified (except one *microwave food*). However, some activity classes are sometimes confused, such as, *stacking objects* and *unstacking objects*, which is somewhat expected given the high visual similarities between these activities. This means the topics b, e and g are likely to assign high probabilities to a set of common code words, and that when a video in one of these classes is represented as a mixture of topics, all three topics are relatively high; classifying the video by selecting only the highest probable topic does not adequately demonstrate the mixture over the topics and can negatively effect the attempted mapping between ground truth labels and emergent topics.

Results: Leeds Activity Dataset

Experiment B1 results for the Leeds Activity dataset are presented using two different sets of key objects. In Table 6.13a we present the results of the cluster metrics obtain when using each of the learning methods after encoding the term-frequency matrix and using the 14 *autonomously learned* key objects in the environment. The results for all learning methods (excluding random clustering) are improved by using the low-pass filtered term-frequency matrix, hence we present results using the $(493 \times 2,876)$ matrix. For multinomial distributions, only the highest topic proportion (> 0.3 probability threshold) is selected, where the "# classified" row specifies the

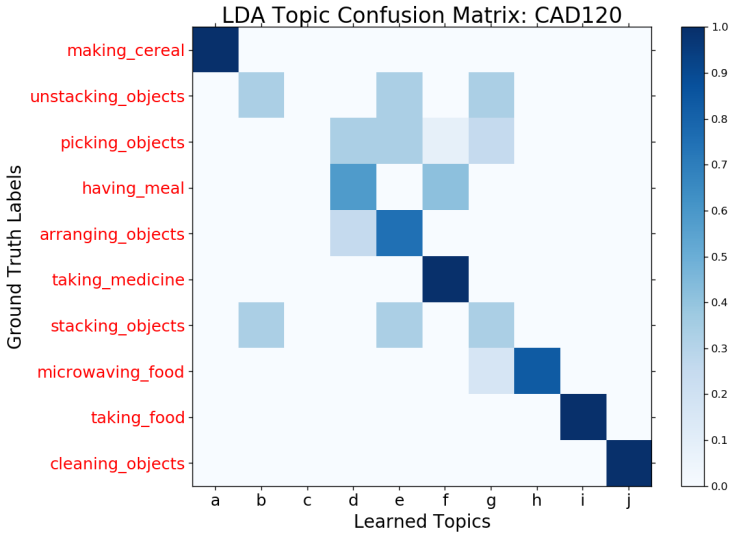


Figure 6.12: Experiment B1: Confusion Matrix for CAD120 dataset: ground truth activity classed vs the 10 emergent LDA topic assignments. Normalised by ground truth labels.

Metric	Random clustering	Unsupervised k-means	Unsupervised LSA	Unsupervised LDA	*Supervised* SVM
# classified	493	493	493	487	493
V-measure	0.05	0.30	0.68	0.63	0.71
Homogeneity Score	0.05	0.24	0.66	0.62	0.71
Completeness Score	0.05	0.39	0.70	0.63	0.71
Mutual Information	0.12	0.31	1.46	1.40	1.59
Normalised MI	0.05	0.54	0.68	0.63	0.71
Accuracy	0.12	N/A	N/A	N/A	0.78

(a) Experiment B1: Results for Leeds Activity dataset using autonomously learned key objects.

Metric	Random clustering	Unsupervised k-means	Unsupervised LSA	Unsupervised LDA	*Supervised* SVM
# classified	493	493	493	486	493
V-measure	0.05	0.41	0.65	0.71	0.73
Homogeneity Score	0.05	0.36	0.63	0.67	0.73
Completeness Score	0.05	0.47	0.66	0.74	0.73
Mutual Information	0.12	0.80	1.42	1.51	1.63
Normalised MI	0.05	0.41	0.64	0.71	0.73
Accuracy	0.12	N/A	N/A	N/A	0.80

(b) Experiment B1: Results for Leeds Activity dataset using SOMa annotated key objects.

Figure 6.13: Experiment B1 results for Leeds Activity dataset using different sets of key objects in the environment: Cluster metrics obtained comparing the ground truth labels of 493 segmented video clips, recorded from a mobile robot and encoded using a qualitative framework, against the learned, emergent human activity classes. The table shows methods of increasing sophistication: unsupervised k-means clustering; low-rank approximate LSA; Generative LDA; compared against random chance and a supervised SVM as an intuitive lower and upper bound respectively.

number of observations classified above this threshold. When the number classified is less than 493, it means that some observations are not sufficiently considered as any specific topic, meaning they were a mixture of many topics, each with low probability.

Similarly, we present results in Table 6.13b when using a set of 12 manually specified SOMa objects in the environment. This results in a $(493 \times 3,543)$ term-frequency matrix (after the low-pass filter is applied). We can see here that manually specifying the key object locations across the environment helps to obtain more coherent clusters of human activity with respect to ground truth labels.

One hypothesis for this improvement is that given more accurate key object locations in an environment, more rich qualitative representation between the human pose and the objects can be encoded, leading to a higher code word frequency for the same video sequence (and a larger code book). However, the slight improvement in results must be balanced against the extra effort involved to manually locate key objects in new environments.

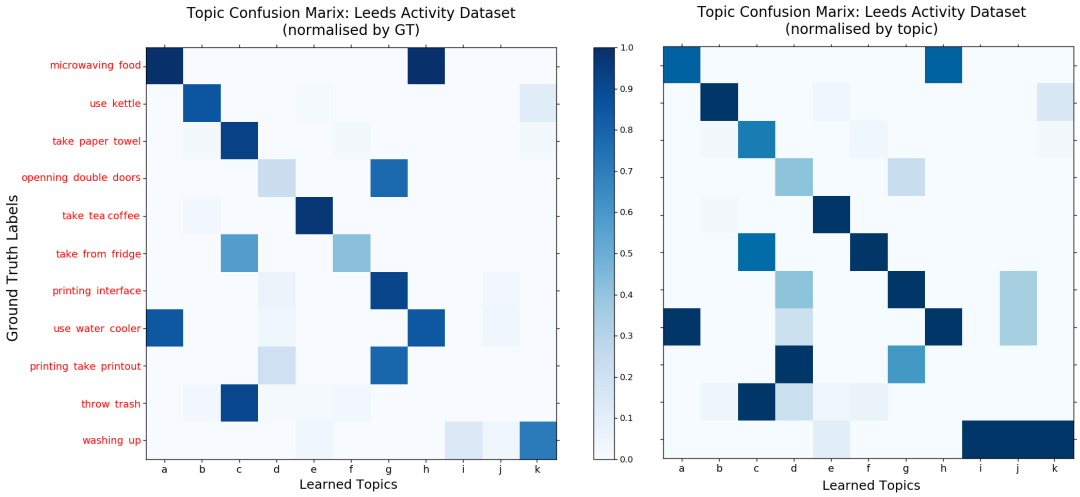


Figure 6.14: Experiment B1: Confusion Matrix for Leeds Activity dataset: ground truth activity classed vs the 11 emergent LDA topic assignments. (left:) Normalised by ground truth labels. (right:) Normalised by topic assignment.

Our proposed learning methods, LSA and generative LDA, significantly out perform the simple unsupervised k-means clustering algorithm and the uniform random assignment. Our interpretation is that these methods better generalise observations than the more simple method, since they consider qualitative features with similar meaning, i.e. identifying synonymy between encoded dimensions, unlike the k-means algorithm. It is interesting to note that when using this more challenging dataset, where the intra-class similarities are much higher that the CAD120 dataset, LDA outperforms LSA when using the SOMa annotated key objects. We believe this is because the LDA model is able to model a mixture in the training samples better, so even in training samples that are difficult to classify, the topic distributions are more representative. Further, it can be seen that the supervised approach obtains 80% accuracy on this challenging real-world dataset, and performs only slightly better than the unsupervised LDA method, when evaluating how well coherent the emergent classes are to the ground truth; even though the SVM has access to labelled training instances to create decision boundaries.

Figure 6.14 presents the classification results for the LDA results as a Confusion Matrix of the ground truth activity labels vs the classified topic labels, both when using SOMa annotated objects. Due to the highly unbalanced activity classes in terms of the number of instances of each, we normalise these matrices by both ground truth labels (left) and by topic assignments (right). From this presentation, we can interpret that the activity class *washing up*, is almost entirely classified into Topic ID 10 (left), but also that Topic ID 8 and 9 consist of mainly these ground truth videos (right). Topic k is a highly frequent topic that represents the human activity, however, topics i and j may represent this activity being performed in a slightly different visual way. This variety is less frequent, as only a small number of videos are classified into i or j. Further, the large imbalance between the number of each activity class observed,

causes some topics to contain multiple activity classes. For example, Topic g can be seen (left) to classify three different activities, *openning double door*, *using printer interface* and *taking the printout*. However, it can be seen (right) when normalising per topic, there are very few instances of *openning double door*, and so the topic distribution is dominated by the remaining two ground truth classes.

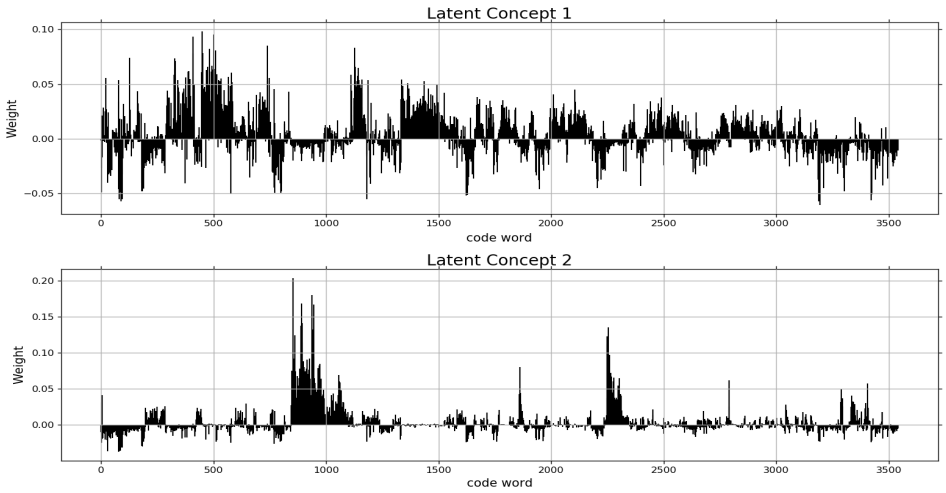
Interpreting the Learned Activities

One of the limitations highlighted regarding the use of unsupervised learning methods on high-dimensional data is that they can often be difficult to interpret. We wish to investigate not only how coherent the emergent activity classes are, by comparing them to the ground truth labels as above, but also investigate what the learned distributions represent; how interpretable are they. Here we present a closer look into the learned activity classes from the Leeds Activity dataset. We show that for LSA and LDA learning methods used in Experiment B1, different graphical techniques can be used to interpret the emergent concepts and topics.

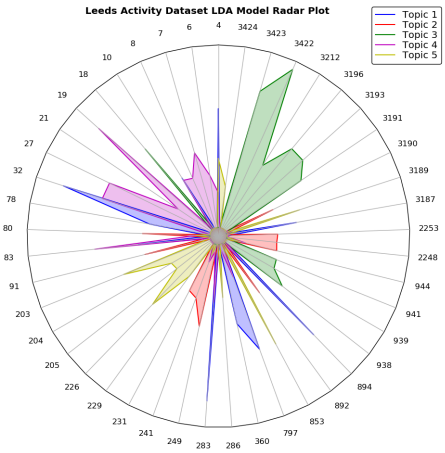
LSA Given the encoded $(493 \times 3,543)$ term-frequency matrix from the Leeds Activity dataset, the LSA decomposition recovers the matrix singular values in the non-increasing diagonal matrix Σ, which represents the weighting of each concept in the matrix. However, to interpret the emergent concepts we are interested in the right singular vector V^T, that specify the rotations around each axis giving us the importance of each code word feature in the vocabulary for each latent concept.

For the Leeds Activity and the recovered 11 concepts in Experiment B1, the $(N \times 11)$ matrix V^T contains an assignment weight for each discrete code word for each of the latent concepts, where $N = |\mathcal{V}_d| = 3,543$. Two of the right singular vectors from the decomposition are shown plotted in Figure 6.15a. We consider these singular vectors as signatures that each represent a learned human activity and present it as a linear combination over the code book vocabulary.

LDA Since LDA topic distributions are full probability distributions and do not contain negative values, we can visualise multiple topics on a single Radar plot. Five multinomial topic distributions, $\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_5$, learned from the generative LDA method in Experiment B1 are plotted in Figure 6.15b over a reduced code book vocabulary of 45 code words. In a Radar plot, the height of a line represents the probability of a specific code word in a topic represented as coloured lines and the axes represent the ID of the code words. The plot is a method of highlighting the specific code words which are common between learned topics and those that are considered discriminatory for each human activity. Here, the most highly probable 10 code words are taken from each of the 5 topic distributions (with 5 common code words meaning 45 axes). It can clearly be seen that the human activities represented by these topics overlap with respect to some code words as expected. However, each topic also has high probabilities for unique code words which are not common between classes, and considered discriminatory features for those topics.



(a) LSA decomposition showing two right singular vectors of length $3,543$, that represent the weighting of each code word in the vocabulary on two latent concepts that represent two different human activities.



(b) Radar Plot showing 5 learned multinomial topic distributions. The hight of the line defines the probability of the code word in the distribution. Only the highest probable code words in these topics are displayed (over a subset of 45 highly probable code words). Best viewed in colour.

Figure 6.15: Experiment B1: Interpreting learned activity classes using LSA and LDA unsupervised methods.

Discussion

The results presented in Experiment B1 demonstrate that a number of emergent activity classes can be recovered from real-world, challenging human observations: first from a static camera set-up with actors performing daily living activities with the presence of dynamic objects, and secondly from a mobile robot observing an unstructured, human populated environment where no restrictions were places on the kinds of activities or dynamic interactions observed. This

mobile robot dataset contains many occlusions, plus difficult and fast-paced human-object interactions. The dataset presents high intra-class variation, which is somewhat shown by multiple viewpoints of the same activities taking place, requiring a view invariant qualitative encoding in order to generalise observations and learn from multiple observations in the qualitative space, instead of the quantitative space.

The results show that emergent topics are learned and that in the majority of observations, the topics are coherent with human annotated ground truth labels. It can be seen that the methods proposed significantly improve upon simple unsupervised clustering and achieve similar performance to a supervised learning approach. This shows the qualitative framework used to abstract the observations are somewhat viewpoint invariant and can handle large amounts of noise and variation during the observational phase, and that unsupervised learning methods are sophisticated enough to separate observations into coherent classes. One conclusion drawn from the comparison of the results across the two datasets, is that LSA seems to perform better with more simple human activity data, with fewer overlapping code words. Whereas LDA seems to perform better when large inter-class similarities and intra-class differences exist. That is, when the observations are taken from real-world and noisy human interactions, the generative probabilistic method seems to be able to better learn patterns of human activity.

A key distinction between the CAD120 dataset and the Leeds Activity dataset is the presence of dynamic objects tracked in the Cornell dataset. It is clear that the granularity of activities learned from the Leeds Activity dataset could be more detailed if smaller, dynamic object locations were available. Further, it could improve the activity learning process since more discriminatory details about the observation could be encoded. For example, a person carrying a cup is likely to perform a different activity than one carrying a piece of paper, however this is not currently detectable by the robot. However, to add this into an embedded robotic system is particularly challenging. For example, objects would need to be recognised and tracked across images where a human body pose is also estimated. To best encode the human pose, the entire person must be contained in the camera frame, whereas, to best detect objects, they should be large and located centrally in the image. Therefore, sufficiently capturing the human pose and dynamic objects at the same time is very challenging. In addition, semantic knowledge such as the abstract object class could be applied to each detected objects. Web mining methods for applying this semantic knowledge have been investigated in [Young et al., 2017], however, this is out of the scope of this thesis.

6.3 Concluding Remarks

To summarise this chapter, we have shown that from multiple human observations in real-world environments, it is possible to learn consistent and meaningful patterns of 2D trajectory motion behaviours and more detailed 3D human body pose sequences using unsupervised learning methods applied to our novel qualitative representation of human observations. The first part

of the chapter focuses on learning human trajectory motion patterns and that once a model is learned, the robot can predict regions on the map that are most likely to be occupied in the future given a newly observed trajectory. The robot is able to make initial classifications in order to make a prediction from just 0.4 seconds worth of observation, generalising very few human trajectory poses. We also showed that the classification of new trajectories into motion patterns improves with more training data available.

In the second half of this chapter we showed that using more detailed human body pose information, multinomial topic distributions over a discrete vocabulary can be learned and shown to align well with human annotated ground truth activity classes. Models of human activities were learned with the presence of dynamic objects in a staged static camera set-up dataset (CAD120), as well as using highly unstructured real-world environments with static objects automatically learned from the robot. We demonstrated that when manually specified key object locations are used, the qualitative approach is better able to capture the dynamic behaviour of human movements. This can be expected, but relies on manual segmentation of new environments. We presented a comparison between our proposed unsupervised methods to a standard supervised SVM method in order to add a perspective to the learning performance. It was shown that the performance of LSA and LDA in these settings is similar to the supervised technique, however, interpreting the model learned is more challenging. Finally, an interesting result was that LSA performs better than LDA when the human observations in the dataset are more staged and more clear (CAD120 dataset), whereas LDA generalises noisy human pose observation better, in the case of the Leeds Activity dataset when using manual object locations.

We have recorded two real-world human datasets recorded from a mobile robot: first a six week human trajectories dataset and secondly, a one week human body pose dataset. These are both available online for the community's use, along with meta data regarding maps and ROS message definitions.

The Leeds Activity dataset was recorded from an unstructured human environment, however, the analysis was not performed entirely "in-the-wild". There remains several practical considerations that limit the effectiveness of the framework in a lifelong robotic deployment setting, such as, how to autonomously obtain human observations, how to deal with continuous streams of video data and employ learning methods that are incremental and more efficient. In the next chapter we discuss some of these issues and attempt to remove some of the assumptions in order to allow for the learning framework to be deployed in a continuous, lifelong learning setting.

Chapter 7

Experiments: Practical Considerations for Lifelong Learning

In this chapter we propose methods to address some of the remaining limitations and assumptions that impede the framework from being truly useful in a continuous, life-long learning settings. Our aim is to build upon our qualitative relational framework introduced in the previous chapters and obtain a more general framework by addressing some of the difficulties and considerations for real-world deployed mobile robots. The assumptions we challenge in this chapter are as follows:

1. The recorded human pose sequences are temporally focussed around a single human activity instance occurring, as opposed to multiple overlapping activities, or similarly, no interesting activity occurring at all.

2. The robot has unlimited time and computing resource to learn human activity classes by repeatedly performing a batch learning process over an ever increasing set of observations, instead of using more efficient, incremental learning methods to build upon its knowledge over long periods of time.

This chapter is spit into these sections, we briefly introduce the practical considerations to address each of these assumptions and propose our approach, before describing an experimental set-up and results used to validate our approach.

Continuous video streams This is the assumption that human observations recorded by the robot consist of a consecutive sequence of images where the body pose of a single person performing an activity can be estimated perfectly, i.e. a person enters the robot's frame of

view, performs a typical human activity in such a way to not occlude any body joints, and then exits the camera frame. In practice, for robots deployed in real-world environments, this does not occur often. The robot's human observations often consist of people performing multiple overlapping activity classes, or performing only part of an activity within the robot's field of view. Other observations may containing no interesting behaviour as the person might not interact with anything in the environment and just walk past the robot.

Given a sequence of images recorded by the robot, it is not ideal or always possible for a human to manually segment the interesting human activities occurring in the sequence. We propose to use Latent Dirichlet Allocation applied to encoded activity histograms obtained using our qualitative framework in order to handle these challenging sequences. We test the hypothesis that LDA models each observation as a mixture of emergent topics, and therefore assumes that multiple activities are occurring in each observation. This allows the robot to learn coherent activity classes even when the video sequences are not temporally segmented into perfect sequences focused on a single activity. In Section 7.1 we provide two experiments with increasing degree of temporal segmentation applied to the recorded observations to demonstrate that this approach works well.

Incremental learning Given an autonomous mobile robot operating for weeks or months at a time, it is not efficient to repeatedly perform batch learning on an ever increasing set of recorded videos, e.g. using techniques such as low-rank approximations (LSA) or Gibbs Sampling (LDA). Therefore, we propose an incremental learning method that can update its learned activity classes based upon only new observations and does not require re-computing for previously analysed data. We propose to use a Variational Bayes (VB) approximation method which aims to optimise a simplified, parametric distribution in order to fit the LDA model posterior using mini batches of observations. Using this method, the robot can save memory by not storing the exact quantitative recordings of previous observations and instead maintain a much lower dimension distribution over the learned activity topics (and the topic distributions themselves). We describe the VB approximate learning method and present an experiment to compare to standard LDA using Gibbs Sampling in Section 7.2.

7.1 Learning from Continuous Video

Given an autonomous mobile robot operating over a long period of time, there would usually not be the availability of a human annotator or a manual mechanism that replies on human input for segmenting "interesting" sequences from the recorded videos. This means the robot requires methods to learn human activity classes from video observations that are not temporally segmented, that is, where a single, perfect example of an activity does not occur in each sequence. In this section we describe how the Latent Dirichlet Allocation generative framework allows for each human observation to be modelled as a mixture of topics using the mixing pro-

portions vector $\boldsymbol{\theta}$. In text mining literature, this is akin to learning the main topics or themes of a corpus of documents, where each document can contain a mixture of words relating to different topics. For example, a text document that discusses *money involved in professional sports* might be considered as a mixture between mainly a *sports* topic and a *finance* topic, where a mixing proportion vector $\boldsymbol{\theta}$ estimates the proportion over all topic distributions. In our setting, the topics relate to emergent human activity classes and therefore the proportions vector describes the mixture of each activity class estimated from an encoded human observation, and where each activity class is represented as a distribution over the set of qualitative code words. Assuming that each observation contains a mixture of topics is crucial and means that the code words in a bag-of-qualitative-words representation can have been generated from different topics allowing the robot to learn activity classes from video sequences that are not manually temporally segmented, i.e. "in the wild".

7.1.1 Implementation Details

In this section, we present two experiments which aim to highlight the probabilistic mixing property of the LDA model, and describe why it makes the manual temporal segmentation of human observations non-essential for the purpose of learning human activity classes. First we briefly discuss the data capture process of the Leeds Activity dataset and then the implementation details regarding the various methods in which the dataset was manually temporally segmented in order to extract "interesting" human interactions. The three segmentation approaches are: full temporal segmentation, where each activity class instance is segmented; no manual temporal segmentation, i.e. entire recorded human body pose sequences; finally, the in-between case, where manually segmented clips are concatenated back together to form sequences of continuous activities for evaluation purposes. We discuss the details of this segmentation next.

Temporal Segmentation

The Leeds Activity dataset is described in 6.2.2, where human body pose sequences and RGB images are recorded using OpenNi2 in real-time to detect the person in the robot's field of view and estimate the body pose. This process repeats for all the images the person is detected by the robot resulting in a sequence of images and pose estimates recorded. Volunteers manually segment the dataset into shorter video clips focussed around a single human performing an activity. This steers the learning methods, shown in the previous chapter, towards representing the repeated qualitative behaviours in these clips as the emergent activity classes. However, when no temporal segmentation is applied there are many different interactions within the video sequences that are repeated or common that were not considered "interesting" activities by the annotators and therefore excluded from the segmented clips. For example, most of the observations will contain code words that may relate to the human body pose when stood idle, or walking. From the 293 human observations recorded in the dataset, 77 contained none of

the ground truth activities (as annotated by volunteers), however they all contain a sequence of estimated human body poses. This is a major challenge when using real-world, non-segmented video sequences recorded "in-the-wild".

Similarly, the repetition of activities in the dataset is one of the reasons the volunteers annotated the sequences with a ground truth label, i.e. the set of labels derives from interesting activities that were commonly repeated. However, other activity classes occur in the recorded sequences but with lower frequency, e.g. there is one observation where a person is *cutting a birthday cake*, however this is a rare human activity and therefore was not included in the list of ground truth classes.

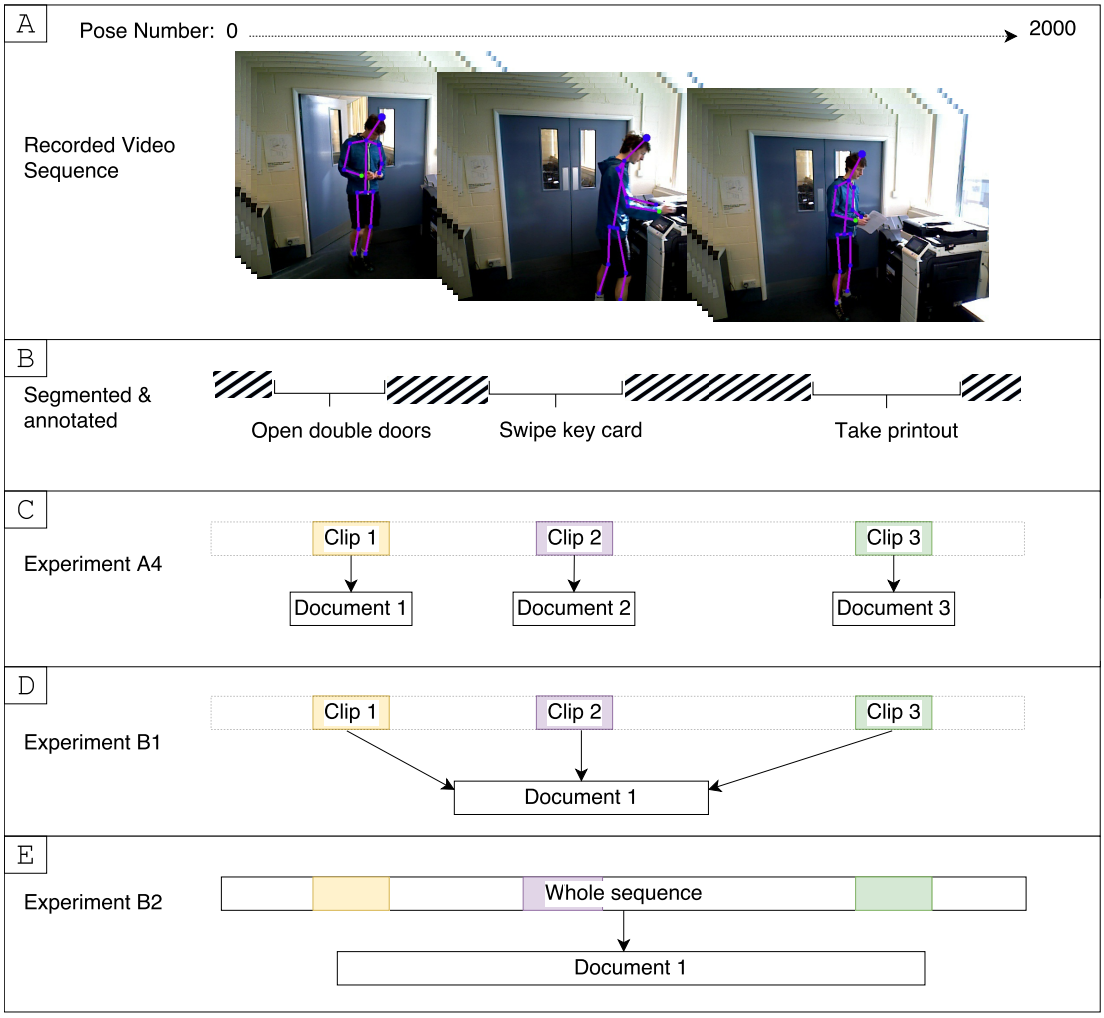


Figure 7.1: Demonstration of three different temporal segmentation methods applied to one recorded video sequence containing three activities as annotated by volunteers. Best viewed in colour.

Using an example, we illustrate the different temporal segmentation methods that are used

in the following experiments.

- Figure 7.1(A) shows an example sequence of RGB images overlaid with human pose estimates representing a recorded observation (comprising of 200 body poses/frames). In this example, a person walks through a set of double doors, towards a printer-copier machine, swipes their key card, collects a printout and walks away from the printer and out the robot's field of view. Nothing particularly unusual occurred in this video sequence.

- Figure 7.1(B) shows the volunteer annotators representation as a sequence of three ground truth human activity instances occurring: first the person *opens double doors*, followed by *swipes key card*, followed by *takes printout*. Each of these annotations includes the temporal sequence of poses/frames where the activity occurs, and the activity class ground truth label. Note there are often large temporal gaps between the annotated sequences, where the volunteers believed no interesting activity is occurring. The excluded sequences, marked with hashes in the image, are extremely difficult to handle for a learning framework because any human behaviour, not restricted to the set of ground truth classes, could be observed.

- Figure 7.1(C) demonstrates how each of these segmented video clips can be considered as their own "document" in text mining terminology and each encoded as an activity histogram over the vocabulary of qualitative code words. This is the segmentation process used in Experiment B1 on the Leeds Activity dataset, resulting in 493 activity histograms each with an associated ground truth activity class label.

- (D) highlights a method to test the hypothesis that LDA models a mixture of activities occurring in each observation. Here, we use the same manual segmentation of the observations, but concatenate the segmented clips back together into a (possibly discontinuous) sequence that excludes the surrounding frames (hashed). For example, a single activity histogram is used to represent the three short sequences of segmented video originating in the same recorded observation. We show in Experiment C1 that the emergent topic distributions relating to activity classes are indeed very similar when using the concatenated sequences versus when using the temporally segmented video clips (as in Experiment B1).

- Finally, (E) demonstrates the most realistic and challenging case where no manual temporal segmentation is applied to the recorded video sequences and thus the sequences are much longer in duration, more noisy and include all surrounding frames that were considered not interesting by the annotators. This translates into encoding much larger interval graphs as both the number of qualitative relations with key objects increases and the temporal duration of the sequences is longer. Learning human activity classes from these sequences is very challenging and we consider this more representative of learning human activities "in-the-wild". In Experiment C2, we frame this as a multi-label problem, since each of the recorded video sequences can contain an arbitrary number of annotated

ground truth activities occurring (along with ground truth labels). We show that the emergent topic distributions when using no temporal segmentation somewhat align with the ground truth classes, although a one-to-one mapping is not expected due to the much more varied video sequences.

7.1.2 Results and Discussion

Experiment C1: Segmented vs Concatenated Videos

This experiment is designed to evaluate the capability of our framework to learn coherent human activity classes based upon video sequences that contain multiple activity instances occurring, i.e. an experiment towards using recorded video with no manual segmentation. The idea is to compare the emergent topic distributions when using the two different segmentation methods. Figure 7.1(D) demonstrates the process by which manually segmented video clips are concatenated back together to achieve a sequence containing multiple activity instances, potentially from multiple activity classes and excluding the surrounding video frames that do not contain "interesting" activities.

The Leeds Activity dataset consists of 287 recorded video sequences manually segmented into 493 annotated clips. We concatenate the clips back together into 210 concatenated sequences (recall that 77 recorded sequences contained none of the ground truth activities). This results in a set of 210 sequences that contain a (mean) average of 2.3 segmented clips per sequence (max = 10), where each clip is associated to a ground truth annotated label, so each concatenated sequence corresponds to multiple ground truth labels.

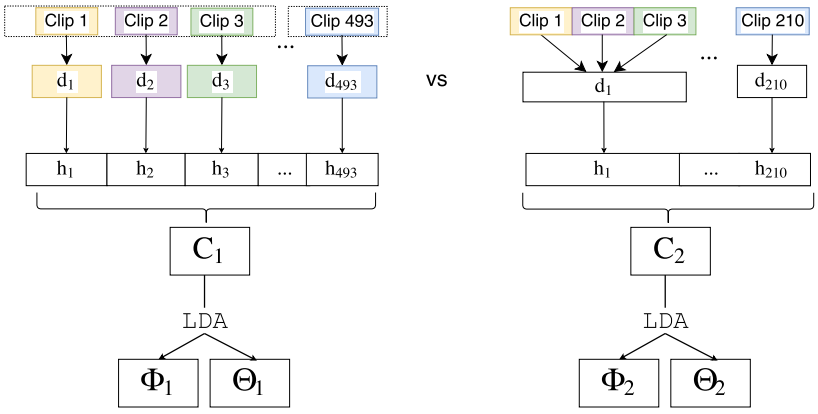


Figure 7.2: Experimental set-up comparing emergent topic distributions when using segmented video clips (Experiment B1) versus using concatenated sequences containing multiple activity instances.

In this experiment it is the same estimated human body pose sequences that are used as in Experiment B1, however multiple activities are now present in each observation (activity

histogram) and so we require the learning framework to be flexible and model each as a mixture of classes. For this reason, we use LDA which estimates a topic mixing distribution vector $\boldsymbol{\theta}$, sampled from a Dirichlet distribution, for each of the input activity histograms in the dataset. This translates as assuming a multinomial distribution over the topics within each observation, removing the requirement for manual temporal segmentation of observations into shorter clips where only a single activity occurs.

We follow the same experimental procedure as in Experiment B1 and the goal here is to compare the learned topic distributions obtained when using the two different segmentation methods. That is, using the temporal segmented clips an $(493 \times 3,543)$ term-frequency matrix C_1 is computed (same as Experiment B1) then, using the concatenated sequences a $(210 \times 3,436)$ term-frequency matrix C_2 is computed. The slight variation in the number of code words is due to the low-pass filter, i.e. a code word must appear in a minimum of 5 activity histograms. The experimental set-up is shown in Figure 7.2. The idea is to compare (Φ_1, Θ_1) with (Φ_2, Θ_2) and evaluate the effects of the extra temporal segmentation applied to videos encoded in C_1. We do this by examining the Cosine Similarity between the two emergent topic distribution matrices Φ_1 and Φ_2, that represents the difference of the angle between the (unit-normalised) topic distributions, denoted by the vectors.

Results

In Figure 7.3 we present a Cosine Similarity matrix between the topic distributions learned using the temporally segmented clips against those learned using the concatenated sequences. We use the Munkres Hungarian algorithm [Munkres, 1957] in order to match the highest corresponding topic distributions together, i.e. an assignment problem. We can see clearly that the learned topic distributions are very similar, even though the input video sequences are segmented differently, i.e. C_2 is encoded using multiple activity classes in each observation. The strong diagonal indicates a good one-to-one mapping between the two recovered topic distributions and demonstrates that the framework presented is able to recover coherent topic distributions representing human activity classes from the observations containing a mixture of activities.

The average Cosine similarity between the two assigned sets of topic distributions is 0.90. This value drops to 0.56 when using the low-rank approximation method LSA to learn the emergent concepts from the two term-frequency matrices. This demonstrates that the probabilistic, generative LDA method is able to better handle human observations that contain a mixture of activities occurring. This is a very desirable property as we move towards using continuous video with no temporal segmentation applied.

Experiment C2: No Temporal Segmentation

This experiment is designed to evaluate the consistency of learned topic distributions with respect to annotated ground truth activity labels when no manual temporal segmentation of the recorded sequences is performed. That is, the robot encodes a bag-of-qualitative-words

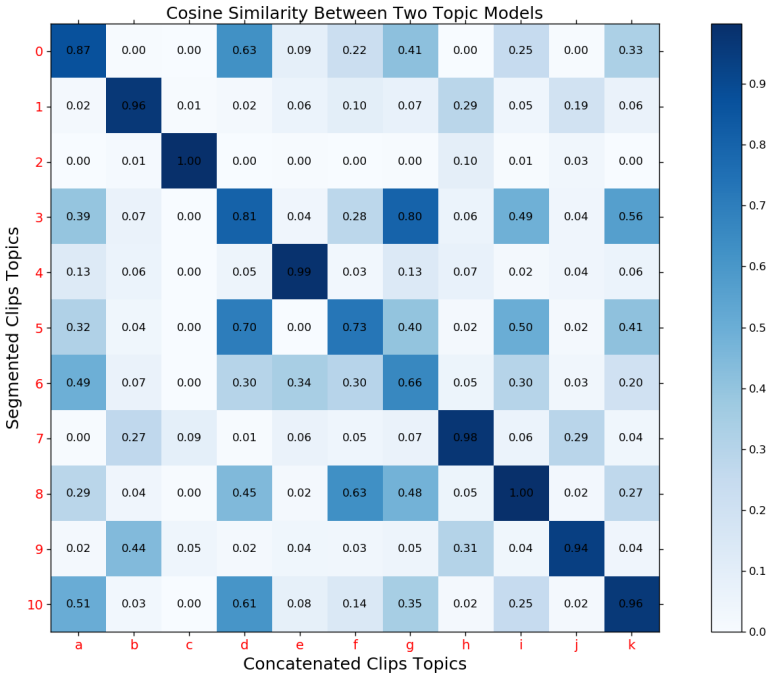


Figure 7.3: Experiment 7.1: Similarity matrix comparing the Cosine Similarity of two topic distribution matrices Φ_1 and Φ_2, where Φ_1 is learned from video clips that have been manually temporally segmented encoded in a term-frequency matrix C_1, and Φ_2 from video sequences containing multiple activity instances, i.e. the segmented videos concatenated back together into discontinuous sequences encoded into C_2.

and a corresponding activity histogram for each complete sequence of recorded human pose estimates it observes whilst "in-the-wild". This is a very challenging task, given the complex dynamics of the environment the robot is required to observe. Figure 7.1(E) illustrates the difference to the previous (manual) segmentation methods, where the sequences here contain all the recorded poses.

All the recorded video sequences in the Leeds Activity dataset are used in this experiment, i.e. $M = 287$, where 77 contain no activity class instance and the full length human pose sequences contain many more interactions, e.g. people walking, standing, chatting. These sequences are much longer with (mean) average: 513 poses and std of 588. They are more varied and contain many human behaviours that were not repeated consistently throughout the dataset, or were not considered "interesting" by the annotators, i.e. those hashed out in Figure 7.1(B). Each sequence can correspond to multiple ground truth activity classes taking place and therefore multiple activity instance labels.

Using the longer and more varied video sequences, there are many more unique code words (graph paths) extracted from the interval graphs. Using the same experimental set-up as Experiment C1 there are $48,172$ unique code words extracted from the observations in the

dataset. In this setting, the code book is much larger and many of the code words represent qualitative relations that are not consistently observed but which are present in at least one observation. For this reason, we experiment using two different low-pass filters, one with the standard filter size of 10 and a second larger filter of 15, resulting in two term-frequency matrices a (287×6521) matrix C_1 and a (287×4531) matrix C_2 respectively. The term-frequency matrices represent each observation as a histogram over the two different code books computed.

For the challenging reasons discussed, we do not expect the learned topic distributions to match one-to-one to the ground truth labels. However, we present a mapping between the learned topics and the annotated labels, which help us understand the consistent topics with respect to the human activities taking place.

Results

Here no manual temporal segmentation was applied to the recorded video sequences and as a result the observations are much longer and considerably more varied. For this reason, there is not a one-to-one mapping present between the learned topic distributions and the annotated activity class labels. To present the consistency, we sum the topic distributions corresponding to each activity histogram which contain each annotated ground truth label (using a low-pass filter > 0.5). This is represented as a matrix and shown in Figure 7.4, using C_1 (left) and C_2 (right).

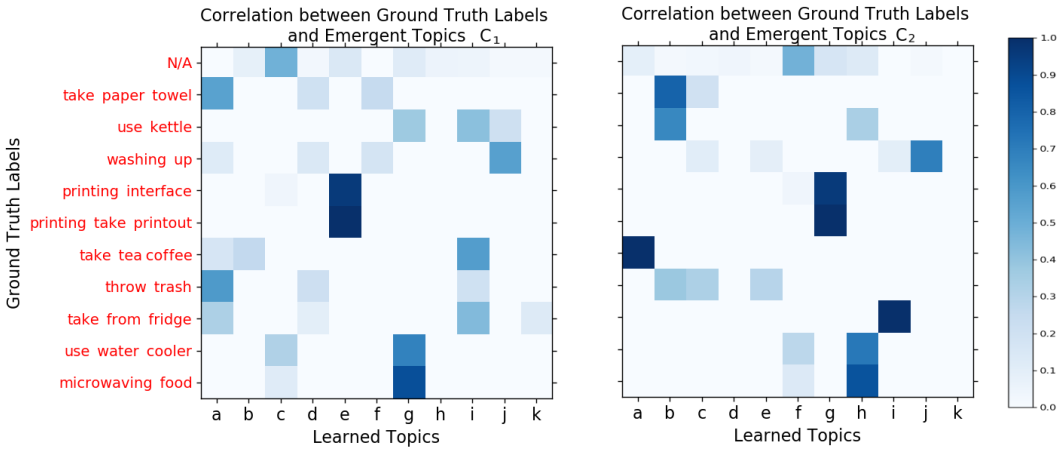


Figure 7.4: Topic distributions learned using no temporal segmentation of video sequences correlating to each annotated ground truth activity class.

We can clearly see (from both matrices) that some activities are not distinguishable from each other, such as the two activity classes that both involve interacting with the printer-copier machine. For brevity, we have removed the "opening double doors" activity class since it has the fewest instances (11) and is not learned. It can be seen that using a smaller code book, the

mapping between learned topics and ground truth activities is more clearly defined. Therefore, we shall discuss the results specific to C_2 (right) from here on. The correlation matrix shows that the majority of the learned topic distributions (columns) correlate to a single or pair of activity class labels (rows). For example topic a correlates highly with "take tea/coffee" and likewise topic j to the activity class of "washing up". Topics such as b relate to a mixture of human activities such as "using kettle" and "taking paper towel". This is intuitive, and based upon the activities that are often observed together (both temporally and spatially), e.g. washing and then drying a mug is a common pattern observed in the dataset when the videos are not segmented.

However, some classes are being confused based upon their spatial arrangement in the environment, for example, the microwave is roughly 30cm away from the water dispenser. These objects are not commonly used together, however topic h contains a mixture of these classes (plus "N/A", and "use kettle"). From manual inspection, we see that people usually stand waiting for the microwave and the water cooler in a similar manner, this is what this topic distribution represents and it is unable to distinguish between the two ground truth activity classes.

Initially, it is unclear what topic d or k relate to, since none of the observed video sequences are classified as these topics above the threshold. However, from manual inspection it is clear that both topics relate to behaviours which occur across many ground truth classes, but below the required threshold. Our hypothesis is that these topics encode common code words that relate to human behaviours exhibited throughout the dataset and not unique to any activity class, such as standing or walking. Thus, the set of "interesting" human activities as defined by annotators could be distinguished using 9 of the 11 topic distributions, if no temporal segmentation is available.

Discussion

In this section, we have proposed LDA as a probabilistic, generative method that can learn topic distributions representing human activities from non-temporally segmented video sequences. That is, removing the manual labour-intensive task of segmenting recorded videos into sub-sequences where only "interesting" activities occur. Clearly, this means that more varied observations are encoded in the term-frequency matrix and the learned topics do correspond directly to the annotated labels.

To summarise the experimental results, we demonstrated that coherent activity classes can be learned from segmented video clips in the previous chapter. This is expanded upon in Experiment C1 where similar topic distributions can be learned when clips are concatenated back together and multiple activities are encoded in each observation. This specifically highlights the mixing property of the probabilistic LDA framework, which estimates a topic mixing proportion vector for each observation. The final experiment here demonstrated that activity classes can be learned from continuous video streams where no manual temporal segmentation

is applied and that these topic distributions are somewhat interpretable with respect to ground truth activity classes, however, a direct one-to-one mapping is unattainable. The analysis of continuous video, given no manual temporal segmentation, leads to the interesting question of "what granularity of human activity constitutes an activity class"?

7.2 Continual Human Activity Learning

To truly integrate in real-world environments, mobile robots with collaborative or assistive human-oriented tasks should be able to continuously learn about their environments and the types of activities that take place in them. From an autonomous robot point of view, this requires incremental learning methods. In this section, we propose a solution to continually learn about human activities based upon incrementally updating the Latent Dirichlet Allocation posterior distribution using Variational Bayes inference (VB) [Hoffman et al., 2010]. This technique was developed to analyse massive corpora containing millions of natural language text documents where batch algorithms were too computationally expensive. It has been shown to converge faster and be as accurate as Markov Chain Monte Carlo (MCMC) sampling methods [Asuncion et al., 2009], and therefore it is ideal for a lifelong learning situation where the number of observations is unknown and could become intractable for batch methods, such as Gibbs Sampling.

The key idea is to incrementally update the topic distribution estimates that represent activity classes of human behaviour. For a new observation the process of updating the topic model is threefold:

1. any new code words in the observations are first appended to the current vocabulary $\mathcal{V}_D$ and to the topic distributions Φ with zero probability,
2. a multinomial distribution over the current set of topics/activities for the new observation is computed, $\boldsymbol{\theta}$, that represents the mixture of topics observed,
3. finally, the topic distributions over the vocabulary (Φ) are updated using this new observation, or mini-batch of observations.

This allows the robot to efficiently update its model of human activities using a single pass over new observations, optimising both storage and computation complexity. Each observation can therefore be maintained as a low-dimensional distribution over the set of topics considered human activities.

7.2.1 Variational Inference for Approximate Activity Classes

The basic idea of variational inference is to formulate the computation of a marginal or conditional probability in terms of an optimization problem. This, generally intractable problem, is then "relaxed", yielding a simplified optimization problem that depends on a number of

free parameters, known as variational parameters. Solving for the variational parameters gives an approximation to the conditional probabilities of interest, in our case, the conditional that defines the LDA posterior.

The posterior distribution in the LDA model is intractable, and so in Section 5.4.1 we introduced standard Collapsed Gibbs Sampling that is an approximate posterior inference technique based upon MCMC sampling independently from the posterior. However, this requires sampling from the entire training corpus meaning the training samples must all be analysed at once. A second category of approximate inference techniques are optimisation approaches. The method we propose to use here is Variational Bayes inference which optimises a simplified parametric distribution based upon the Kullback-Leibler divergence to the posterior [Hoffman et al., 2010, Blei et al.]. The online VB algorithm is provided in Appendix C.5. This technique iterates between analysing a mini-batch of observations and updating dataset-wide parameters. This is particularly relevant to fitting human activity topic models from an autonomous robot "in-the-wild", since it may obtain an intractable quantity of observations to perform standard MCMC sampling approaches. Using this method, the robot can save memory by not storing the exact quantitative observations and instead maintain a lower dimension distribution over the learned topics (and the topic distributions themselves).

7.2.2 Experimental Results and Discussion

Experiment D1: Incremental Learning

Here we repeat Experiment B1 and learn human activity classes from the challenging Leeds Activity dataset where the aim is to validate the use of the incremental VB learning method presented above. The idea is to compare the activity classes obtained when using the more efficient incremental approach, against the standard approach for fitting the LDA posterior using Collapsed Gibbs Sampling which requires multiple passes over all the training samples in order to converge on topic distributions. The same experimental set-up as Experiment B1 is used, the 12 SOMa key objects are used and we encoded the activity histograms using the same qualitative representation. This produced a $(493 \times 3,594)$ term-frequency matrix C, where each has an associated ground truth label from the set of 11 ground truth activity classes as assigned by volunteers. So far, the same as Experiment B1.

Given the limited size of the dataset, we seed the activity model by learning topics using Collapsed Gibbs Sampling [Gelman et al., 2014] on a batch of observations representing the first of observations (day 1) (using $\alpha = 0.005$ and $\beta = 0.01$). This equates to the first 146 observations. Then incrementally add new activity histograms using Variational Bayes with a regular mini-batch size of 5 observations to allow for frequent updating of the topic distributions. To pick the number of topic distributions, we employ a simple method that starts with the number of key objects in the environment, i.e. 12 in this case, and increase by one each day to allow new activities to be learned over time. We also remove any topic distribution that are

not used sufficiently in order to maintain a reasonable number of topics.

Results

Table 7.1 presents results of our incremental, unsupervised concept extraction when compared against ground truth classes. We use the most likely component in a mixture as a label since the proportions vector is multinomial. The method converges to 13 emergent activity classes from the real-world dataset with challenging unstructured behaviour, varying view points, lighting conditions and occlusions. The results show the majority of the instances observed are successfully clustered into consistent activity classes using both the VB algorithm and Collapsed Gibbs Sampling. As an upper bound, we also show the V-measure results obtained when using a supervised (linear) SVM (with 4-fold cv) which has access to the ground truth labels during training; this marginally outperforms the unsupervised techniques, as it did in Experiment B1.

Metric	Standard LDA	Incremental VB LDA	Supervised SVM
# classified	486	493	493
V-measure	0.71	0.66	0.73
Homogeneity Score	0.67	0.63	0.73
Completeness Score	0.74	0.70	0.73
Mutual Information	1.51	1.42	1.63
Normalised MI	0.71	0.66	0.73
Accuracy	N/A	N/A	0.80

Table 7.1: Experiment D1: Comparison between the standard LDA model using Collapsed Gibbs Sampling to fit the topic distributions versus the incremental approach using Variational Bayes inference.

Sensitivity Analysis

The performance of the incremental LDA method was evaluated using the Dirichlet hyperparameters set to $\alpha = 1.5$ and $\beta = 0.65$. In order to evaluate the effect of altering the Dirichlet hyperparameters on the performance of the incremental LDA, we perform a sensitivity analysis on α and β, the parameters that control the prior Dirichlet distributions applied to the sampled per-histogram topic distributions and the topic distributions themselves respectively. This is often referred to as "what-if" analysis, i.e. what would the performance of the incremental learning be using different prior distribution hyperparameters α and β.

Figure 7.5 shows the evolution of the $v - measure$, a standard measure of how well the observations have been clustered, over a grid of values for α (left) and β (right). The optimal value of the other hyperparameter is used when performing this analysis, i.e. when altering α, β is set to 0.65. We can see that values close to 1.5 are optimal for the hyperparameter affecting

the per-activity-histogram topic distributions (α), the distribution over the number of topics. This distribution has relatively small dimensionality and this value has the effect of sampling a small number of the topics to represent each activity histogram observation. Values of β close to 0.6 are optimal in this incremental learning setting for the hyperparameter affecting the topic (or human activity) distributions defined over the code book vocabulary. This is a high dimensional distributions meaning that a relatively small number of code words are selected as important for each topic or activity. Finally, the sensitivity analysis performed in this section agrees with the analysis performed in [Asuncion et al., 2009], which suggests that when using VB to incrementally fit LDA topic distributions, there is a requirement for more smoothing in order to match the performance of batch algorithms.

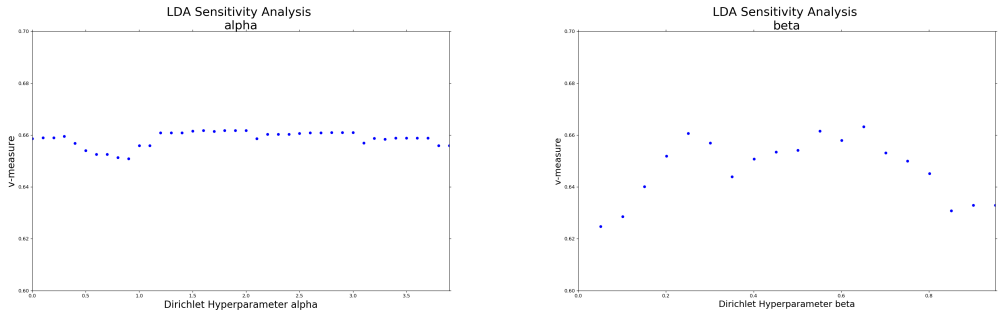


Figure 7.5: Sensitivity Analysis of Dirichlet Hyperparameters, α (left) and β (right) on the incremental VB fitting of LDA for human activities.

Discussion

We have shown that incremental learning methods are possible for complex Bayesian models, where computing the posterior is intractable and only estimated methods are possible. They are particularly useful when learning is performed on long-term deployed mobile robots since there is no way of obtaining all observations in advance, or even knowing the size of the future corpus. However, as shown in Experiment D1, with limited observations the model may not fully converge. Here, we believe that the posterior distribution did not fully converge due to lack of observations. Another feature that can negatively effect the performance of incremental methods is the specific order in which the human activities are observed. As discussed in Section 3.1, the level of abstraction of human activities a learning method can hope to separate is effected by the order the observations perceived. Ideally, the robot would observe a random assignment of different activity classes in order to clearly separate the topic distributions early in the life-long learning process. However in practice, i.e. when the robot is observing the real-world and often static during the recording process, many consecutive observations are recorded of a similar activity class, before new locations were observed. This is a major challenge for incremental learning approaches. Also, the batch methods are able to analyse the least observed

code words and remove them using a low-pass filter. This allows the learning to focus on the most descriptive qualitative code words in the entire dataset. However, using the incremental approach, there is no way of knowing if a code word will be observed any more or less than another. As a result the code book $\mathcal{V}_D$ in Experiment D1 grew to a less manageable $22,829$ unique code words.

A possible future direction of research could be to extend this to many months of observational data. This would allow for totally new topics to be discovered, possibly from the robot incurring new environments. A "learning-rate" could be altered given new environments in order to more quickly converge on new human activities being observed and any topics removed, or not updated, could be considered as the robot "forgetting" a particular human activity.

7.3 Concluding Remarks

To summarise the chapter, we have demonstrated solutions to interesting and as yet unsolved problems in the field of human activity analysis from a mobile robot. We have shown that by using more sophisticated learning methods, it is possible to address some of the practical limitations surrounding lifelong human activity learning from a mobile robot.

Firstly, the LDA assumption that each observation is modelled as a mixture of latent topics allows the robot to learn activity classes from data that is not temporally segmented by humans in advance. Secondly, we have shown that incremental Variational Bayes approximate learning methods can be used to update the posterior distribution of the topic model in order to perform life-long learning, where the number of observations may grow beyond the computing resources available using batch methods. There remains other practical considerations to the work, especially regarding understanding exactly what each topic or code word represents in a large topic model representing human activities. In the next chapter we present the final discussions and draw conclusions from the work presented.

Chapter 8

Discussion & Conclusions

This thesis has presented a novel, qualitative framework for unsupervised learning of human motion behaviours and simple activities for an embedded mobile robotic system. We have demonstrated that activities performed in real-human environments can be learned from long-term observation with minimal expert supervision. A major challenge is the limited field of view of the robot, restricted by the limitations of its sensory modalities, and the complexity of the human populated environments, where observations are often fast paced, obscured or complex. These particular challenges motivate the main contribution of this work of using an abstract, view-invariant qualitative representation coupled with a probabilistic generative learning framework.

This chapter is organised as follows: first we summarise the main contributions of our work, then discuss potential future research that could address some of the current assumptions and challenges. We conclude the thesis with some final remarks about the research field.

8.1 Contributions

The key novelty and contribution of this thesis is a qualitative framework for learning human motion behaviours and activities from a mobile robot's limited view of the world with minimal supervision. This approach is able to utilise incomplete and noisy human observations and generalise them using view-invariant qualitative abstractions in order to extract a set of latent classes defined as human activities. This research makes the following main contributions:

1. A qualitative spatial-temporal vector space encoding of human activities as observed by an autonomous mobile robot.

2. Reliable methods for learning a low-dimensional representation of common and repeated patterns from multiple encoded visual observations. These methods have been shown to extend to cases where no manual temporal segmentation is given, and scale to life-long learning settings using incremental inference techniques.

8.2 Summary

Qualitative Abstraction

To the best of our knowledge, we are the first to combine a generative, probabilistic learning approach, such as Latent Dirichlet Allocation, with a qualitative spatial representation of visual scenes in order to learn real-world human activities. Our novel representation applied to a mobile robot's observations is based upon an object-centric and qualitative abstraction process of video data. This partially alleviates problems associated with low-level image features that have been used with probabilistic learning approaches in the literature, such as in [Niebles and Fei-Fei, 2007]. That is, our qualitative representation maintains semantically meaningful QSR sequences and information specifically relating to the human body pose. Qualitative features that are extracted from an interval graph representation of an observed video sequence arguably encode more "longer term temporal information" than used in similar works. When coupled with the common bag-of-words representation (where word ordering is often lost with image features), the temporal overlap within the code words maintains important structure in the observations while offering the full benefit of sparse, discrete representations. This means our bag-of-qualitative-words representation is able to maintain discriminative qualitative features from visual observations which are used to learn common patterns.

A similar object-based and qualitative abstraction methods has been used in [Sridhar et al., 2010], where the granularity of activities learned in an unsupervised setting are restricted by the perception challenges, as is the case in our work. However, this work uses a single static camera location and does not handle the variability of a mobile robot's environment. An egocentric camera is used to learn a similar qualitative representation of human body pose movements in [Behera et al., 2012b], however this is performed in an offline and batch learning setting that does not address the practical considerations relevant to mobile robotics which is the focus of this work.

Activity Learning

In this thesis, we have demonstrated that simple unsupervised clustering techniques can cluster visual observations encoded using qualitative calculi. However, this method can struggle with the "curse of dimensionality". Alternatively, Latent Semantic Analysis provides a low-rank approximation to the encoded observations by removing dimensions with the least variability. It has been shown to improve upon simple clustering and provide a representation of each video sequence as a linear combination of latent concepts in a term-frequency matrix, as opposed to a deterministic cluster assignment. However, the learned latent concepts are linear combinations of orthogonal feature vectors, which restricts sharing of qualitative features between activity concepts. Finally, we investigate Latent Dirichlet Allocation, which is a generative probabilistic topic model for representing discrete corpora. It represents each video sequence as a probabilistic mixture of topics, and each topic as a mixture of qualitative code words. Since much of

the structure is latent, the multinomial distributions are inferred using sampling or variational inference methods, which are shown to fit topics to human activities consistently. We have also demonstrated the use of variational inference that allows for topic distributions to be fitted in an online and incremental setting, which formulates human activity analysis as a life-long robot learning problem.

8.3 Assumptions and Future Work

In this section, we present some possible directions where our framework could be improved, and highlight where a particular issue has been bypassed. We propose possible future research that could be undertaken in order to make progress with each of these assumptions.

Viewpoint Selection

This relates to the assumption that an autonomous mobile robot knows where to be located in an environment in order to observe and best perceive human activities. Our deployed mobile robot has a pre-built 2D map representation of its environment, but no pre-given knowledge of where humans might perform activities of interest within this environment, or where best to observe them from. Given the limited field of view of the mobile robot's sensors, this information would need to be manually entered, for example, by selecting a specific topological waypoint and camera orientation for each interesting region to observe.

As part of the STRANDS system we have integrated a heuristic method to randomly generate locations within the map to observe human activity[1]. The robot then attempts to obtain a person's physical consent to store their personal data (their image), by proposing options on the robot's screen. Random poses are generated as navigation goals based upon the location of autonomously learned key objects in the environment, and a region around the objects is auto-generated from a distance heuristic. A STRANDS robot used this method for 3 months during a long-term deployment. Ideally, one would investigate how "good" the human recordings from each random location were, where a good view point could result in a long detection, or a detection that is well classified into a learned activity class. This would allow the robot to learn a distribution over the 2D map to represent good future view point locations to observe human activity.

Object Representations

In future work, one would ideally increase the area of the learning environment in order to observe more repeated visual structure. For example, allow the mobile robot to patrol more than one kitchen or student area in order to observe similar objects or human activities performed at different locations across the environment. To scale to these environments, a more abstract

[1] https://github.com/PDuckworth/activity_analysis

object representation would be required, such as learning about object affordances or a high level classification of common objects. This abstraction can be automatically learned from observations, e.g. [Sridhar et al., 2008], or an object label hierarchy can be learned by querying a database, e.g. [Young et al., 2017]. This may facilitate the transfer of learned human activities into new regions based upon similar object abstractions being present in the scene.

Another interesting research direction is to remove object information, such as object labels or class, from the learned activity distributions. This may allow a robot to learn general, object-independent actions or activities, such as "picking" and 'carrying" that can be applied to many different objects.

Qualitative Representation

In this thesis, the aim is to develop an incremental learning framework. For this reason the qualitative spatial abstractions the robot uses to encode video sequences are manually defined in advance in order to discriminate the types of expected human observations. However, if batch learning was appropriate, calculi can be optimised to the specific observations and a learned qualitative representation may better represent the corpus of video sequences, such as in [Behera et al., 2012a].

A future research direction is to incrementally learn a qualitative representation of visual observations at the same time as using this representation to learn concepts, such as sequences of qualitative relations that relate to simple human actions. I co-authored recent work [Alomari et al., 2017b], that attempts to address this problem using an Incremental Gaussian Mixture Model where a representation of separate feature spaces evolves with more observations, starting with an unspecified number of concepts.

Exchangeability Assumption

The proposed bag-of-qualitative-words framework relies upon a simple *exchangeability* property of observed code words within an observation and unordered video observations. That is, de Finetti's Theorem of exchangeability [de Finetti, 1990] states that unordered or exchangeable random variables can be modelled via a latent random mixture. However, it may not be appropriate to model observations of human behaviours or observed code words as unordered or exchangeable. For example, certain human activities or spatial relations could by more likely to be observed following others. One approach in the literature that relaxes this assumption when modelling natural language corpora uses a hierarchical generative probabilistic model and the notion of word order [Wallach, 2006]. Here, each topic is represented as a set of distributions and can be used to predict the "next" word to appear. This is an obvious extension to our work and would allow a mobile robot to predict future states of human interactions.

Improved Datasets

Future research also includes recording a greater quantity and variety of human observations, that is, improving the observational data the robot can learn from. This could be achieved by:

1. Increasing the number of video sequences observed. Ideally the robot would obtain more observational sequences over an extended time period, but where the observations themselves are similar in length to those obtained in this work. Observing human activities over a much longer duration could facilitate learning time dynamic topic distributions which can evolve over time, given that the underlying patterns of observed activities evolve. For example, during different semesters at a University, the human populated environment varies with, for example, the presence off undergraduate students in common areas. Topics that evolve over time has been investigated for natural language corpora using Dynamic Topic Models [Blei and Lafferty, 2006], where topics have been found to evolve across multiple years of Science publications dependent upon the scientific community. The topic distributions are updated with batches of new documents (each year) and the evolution of highly probable topics or code words can be traced through time.

2. Secondly, longer individual sequences of human observations could be recorded. These sequences could contain more information about the transitions between human activities and contain more instances of complete activities taking place in a particular domain. This could facilitate the learning of an activity hierarchy. However, this is very challenging from an autonomous mobile robot's point of view.

3. Using more sophisticated visual object-based abstraction methods in order to obtain more detailed and varied observations of human activities containing hierarchical structure and also the possibility of including smaller, dynamic objects into the activity classes. Learning a hierarchy in this way could relate to the abstract notion of *activity granularity*, where different levels of the hierarchy could be associated with different object abstractions that may relate to different granularities of human activities. Understanding human activities at multiple granularities is interesting future research.

Further Impact

Ideally our framework for unsupervised learning of observed human activities can be coupled with robotic control in order for a robot to physically assist in an activity it observes, or predicts will occur. For this purpose, mobile robotics is clearly an interesting and promising application area for this work. However, another real-world application where a positive impact could be achieved is in the area of intelligent wheelchairs. The observational data from wheelchairs could be used to learn human behaviours of the users over time and prompt highly probable activities for easier control and better quality of life. This data-driven, online learning for wheelchair prompting is something that is currently lacking in intelligent wheelchairs and recommended as a potential benefit to users in the literature [Viswanathan et al., 2013].

8.4 Final Comments

In summary, we have introduced a novel, low-dimension representation of human activities based upon first abstracting qualitative spatial relations between tracked objects in a visual scene, and secondly using unsupervised learning techniques to represent a corpus of robot observations as efficient low-dimensional topic distributions. We have provided a formal representation of human observations as acquired by the robot, qualitative abstractions to generalise these and a method to extract discrete features as sequences of observed semantic qualitative relations. Different methods have been investigated to learn low-dimensional representations of human activities in an unsupervised setting along with experiments and results to validate our approach. Lastly, the framework has been shown to work well given practical challenges of mobile robotics less often reported on.

Open source software has been developed[2], and two long-term mobile robot datasets[3] have been made accessible for the community's use. Therefore, it is our hope that the work presented in this thesis will somewhat help human activity analysis researchers move away from standard offline approaches applied to static, pre-processed visual datasets, and other solutions, such as ours, can be developed to generalise to real-world and varied environments that mobile robots actually inhabit. These solutions are more practical for the evolution of mobile robotics research and benefit the community in the long term.

[2] qsrlib.readthedocs.org

[3] Trajectories: http://doi.org/10.5518/34 and Human Body Poses: http://doi.org/10.5518/86.

Appendices

Appendix A

ROS Implementations

ROS uses a simplified messages description language for describing data values that ROS nodes use to communicate, these descriptions are known as ROS *messages* [ROS msgs, 2017]. Custom ROS message descriptions, as opposed to pre-built standard descriptions, are stored in *.msg* files. A description of the custom ROS messages designed for the techniques implemented on the robot are given here.

ROS message definitions are populated by a set of fields, where each field has a 'type' and a 'name', i.e. *fieldtype1 fieldname1*. The 'type' can be based upon either a standard ROS message, such as 'String' or 'Pose' (pre-built ROS messages available in the geometry package), or another custom ROS message. If a field is comprised of a collection of another message definition it is implemented using a list, e.g. *fieldtype2*[ ] *fieldname2* where '[ ]' defines a list of 'fieldtype2' messages with name 'fieldname2'.

A.1 Human Trajectory ROS msgs

A human trajectory is implemented as a custom *trajectory.msg* message and can be seen in Figure A.1 (left). The key parts of this message are the (x, y) coordinates of the observed detection and implemented as a list of standard PoseStamped messages and called 'Trajectory'; the robot's pose at each detection as a list of standard Pose messages; the metric length of the trajectory is a floating point number; the total displacement (between the first and last pose) of the trajectory in metres; and its start/end time stamps as standard ROS time messages. A collection (or list) of multiple custom trajectory messages are defined as a custom *trajectories.msg* definition which can be seen in Figure A.1 (right).

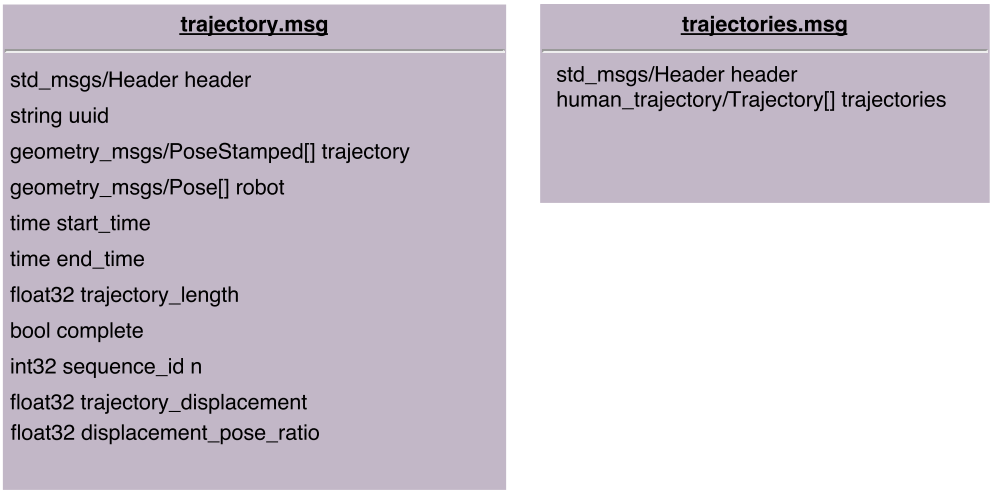


Figure A.1: Custom ROS message definitions for a *trajectory.msg* that represents one observed human trajectory pose, and a collection of such observations, represented as a *trajectories.msg*.

A.2 Human Pose Sequence ROS msgs

The observed human pose sequences are also implemented as custom ROS message definitions. For this task, we developed multiple custom messages to capture the quantitative estimates of each of the detailed human body joints, which make up the human body pose, per observed video frame or timepoint.

The four custom message definitions can be seen in Figure A.2 and include: *joint_message.msg* for capturing the (x, y, z) of a single human body joint as a standard Pose msg; *robot_message.msg* to capture the robot's position in the map coordinate frame, along with its pan-tilt unit angles, (used to translate the joint poses into map coordinate frame); *human_pose_message.msg* represents a single human pose estimate, comprising of 15 body joint msgs, and a unique identifier (uuid) passed from the OpenNI tracker; and finally *human_sequence.msg* which accumulates the time series of both the human pose estimates and the robot pose messages given an observation. The human sequence message also contains information about the time and location of the detection, as well as simple statistics such as number of poses in the sequence etc.

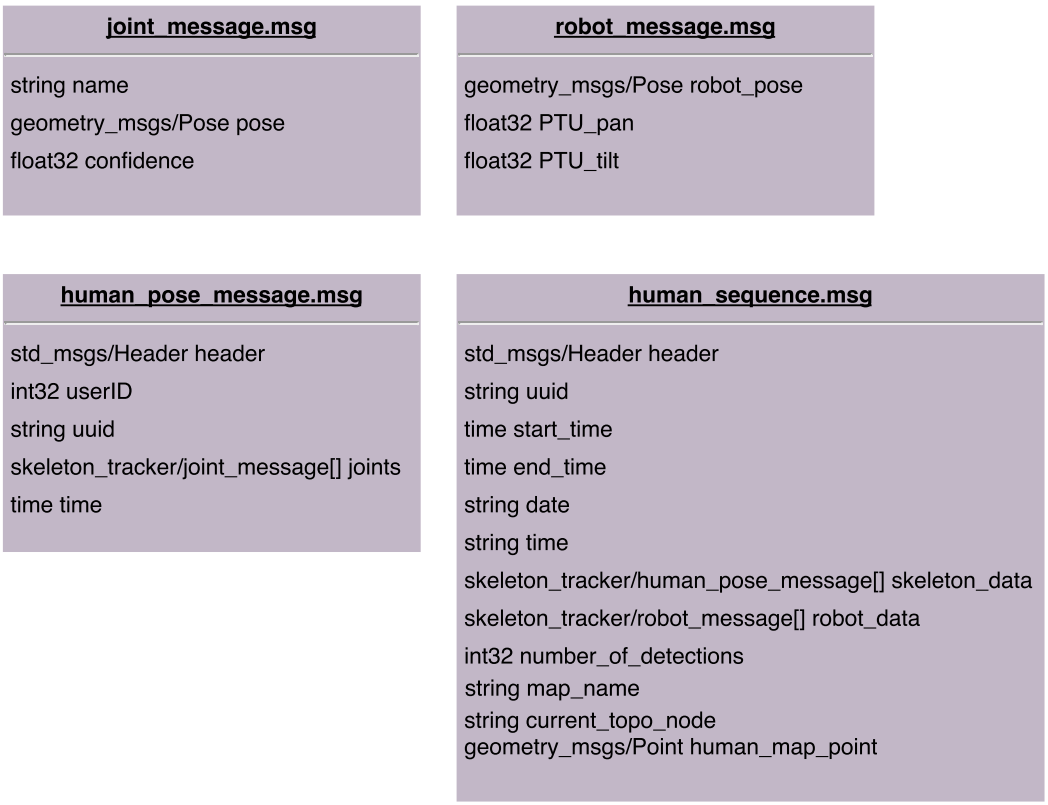


Figure A.2: Custom ROS message definitions used to store the quantitative data of a human observation. (top-left:) A single body joint pose (joint_message.msg), (top-right:) a single robot pose (robot_message.msg), (bottom-left:) a human pose estimate (human_pose_message.msg) and (bottom-right:) a human body pose sequence (human_sequence.msg).

Appendix B

Linear Algebra

B.1 Matrix Concepts

Some general concepts to handle matrices, used throughout this thesis:

- If $\boldsymbol{X}$ is an $m \times n$ matrix and $\boldsymbol{Y}$ is an $n \times p$ matrix, then $\boldsymbol{Z} = \boldsymbol{XY}$ is an $m \times p$ by matrix multiplication.
- Matrix multiplication is associative, i.e. $\boldsymbol{A}(\boldsymbol{BC}) = (\boldsymbol{AB})\boldsymbol{C}$.
- The Identity matrix $\boldsymbol{I}$ is a square matrix with 1 in the diagonals and 0 elsewhere. That is, $\boldsymbol{M}_{m\times n}\boldsymbol{I}_{\boldsymbol{n}\times\boldsymbol{n}} = \boldsymbol{M}$.
- $\boldsymbol{M}^{-1}$ is the inverse of $\boldsymbol{M}$ if $\boldsymbol{M}\boldsymbol{M}^{-\mathbf{1}} = \boldsymbol{I}$.

B.2 Rank and Orthogonal Bases

A collection of vector and matrix calculus:

- A set of vectors $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ is linearly independent if no vector v_i can be expressed as a weighted combination of the other vectors $v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_n$.
- An $m \times n$ matrix $\boldsymbol{M}$ has rank r, where $r \leq \min(m, n)$, if r is the size of the largest set of linearly independent row (or column) vectors of $\boldsymbol{M}$.
- Two vectors $\boldsymbol{v}, \boldsymbol{w}$ of the same length n are orthogonal if $\boldsymbol{v} \cdot \boldsymbol{w} = 0$.
- $\boldsymbol{v}$ and $\boldsymbol{w}$ are orthonormal if in addition they are unit vectors.
- If $\mathcal{V}$ is a set of n orthonormal vectors $\{v_1, v_2, \ldots, v_n\}$, and each vector is also of length n, then $\mathcal{V}$ is an orthonormal basis.
- An orthogonal matrix is one in which the columns (or rows) form an orthonormal basis.

Appendix C

Probability and Sampling

C.1 Dirichlet Distribution

The Dirichlet Distribution pdf with parameter $\boldsymbol{\alpha}$ is given by:

$$p(\boldsymbol{x}|\boldsymbol{\alpha}) = \frac{\Gamma(\sum_{i=1}^{d} \alpha_i)}{\prod_{i=1}^{d} \Gamma(\alpha_i)} \prod_{i=1}^{d} x_i^{\alpha_i - 1}, \tag{C.1}$$

where Γ is the gamma function and for "observations" $\sum_{i=1}^{d} x_i = 1$ and where $x_i \geq 0$.

C.2 Gamma Distribution

The Gamma Distribution pdf with a shape parameter α and a scale parameter β is given by:

$$p(\boldsymbol{x}|\alpha, \beta) = \frac{1}{\Gamma(\alpha)\beta^{\alpha}} x^{\alpha - 1} e^{-\frac{x}{\beta}}. \tag{C.2}$$

C.3 Gamma Function

The gamma function is an extension of the factorial function, with its argument shifted down by 1. That is, if n is a positive integer:

$$\Gamma(n) = (n-1)! \tag{C.3}$$

C.4 Multinomial Distribution

Multinomial Distribution for $x_i \in \{0, \dots n\}$ is given by the pdf:

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = \frac{n!}{\prod_{i=1}^{d} x_i!} \prod_{i=1}^{d} \theta_i^{x_i}(\alpha), \tag{C.4}$$

for $n = \sum_{i=1}^{d} x_i$, $\sum_{i=1}^{d} \theta_i = 1$ and $\theta_i \geq 0$.

C.5 Online Variational Inference

An online variational inference algorithm for fitting $\boldsymbol{\lambda}$, the parameter to the variational posterioir over the topic distributions $\boldsymbol{\beta}$. Using per-document variational parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\phi}$, let $\gamma(n_d, \boldsymbol{\lambda})$ and $\phi(n_d, \boldsymbol{\lambda})$ be the values of γ_d and ϕ_d produced by an E-step. The goal is to set $\boldsymbol{\lambda}$ to maximise the Evidence Lower BOund (ELBO) $\mathcal{L}$:

$$\mathcal{L}(\boldsymbol{n}, \boldsymbol{\lambda}) \triangleq \sum_{d} l(n_d, \gamma(n_d, \boldsymbol{\lambda}), \phi(n_d, \boldsymbol{\lambda}), \boldsymbol{\lambda}), \tag{C.5}$$

where $l(n_d, \gamma_d \phi_d, \boldsymbol{\lambda})$ is the d^{th} document's contribution to the variational bound.

Online LDA is described in Algorithm 2. As the t^{th} vector of word counts n_t is observed, the E-step is performed to find locally optimal values of γ_t and ϕ_t, holding $\boldsymbol{\lambda}$ fixed. Then $\tilde{\boldsymbol{\lambda}}$ is computed, the setting of $\boldsymbol{\lambda}$ would be optimal (given ϕ_t) if the entire corpus consisted of the single document n_t repeated D times. $\boldsymbol{\lambda}$ is then updated using a weighted average of its previous value and the $\tilde{\boldsymbol{\lambda}}$, based upon $\rho_t \triangleq (\tau_0 + t)^{-\kappa}$, where $\kappa \in (0.5, 1]$ controls the rate at which old values of $\tilde{\boldsymbol{\lambda}}$ are forgotten and $\tau_0 \geq 0$ slows fown the early iterations of the algorithm. The notation and algorithm is taken from [Hoffman et al., 2010].

Algorithm 2 Online variational Bayes for LDA

procedure
Define $\rho_t \triangleq (\tau_0 + t)^{-\kappa}$
Initialise λ randomly.
for $t = 0$ to ∞ **do**
    E step:
    Initialise $\gamma_{tk} = 1$
    repeat
        Set $\phi_{twk} \propto \exp\{\mathbb{E}_q[\log \beta_{kw}]\}$
        Set $\gamma_{tk} = \alpha + \sum_w \phi_{twk} n_{tw}$
    until $\frac{1}{K} \sum_k |\text{change in} \gamma_{tk}| < 0.00001$
    M step:
    Compute $\tilde{\lambda}_{kw} = \eta + D n_{tw} \phi_{twk}$
    Set $\boldsymbol{\lambda} = (1 - \rho_t)\boldsymbol{\lambda} + \rho_t \tilde{\lambda}$
end for

C.6 Mutual Information Metric

Assume two label assignments (of the same N objects) U and V. Their entropy is the amount of uncertainty for a partition set defined by:

$$H(U) = \sum_{i=1}^{|U|} P(i)log(P(i))$$

where $P(i) = |U_i|/N$ is the probability that an object picked at random from U falls into class U_i. Likewise for V:

$$H(V) = \sum_{j=1}^{|V|} P'(j)log(P'(j))$$

and $P'(j) = |V_j|/N$.

The Mutual Information (MI) between sets U and V is calculated by:

$$MI(U,V) = \sum_{i=1}^{|U|}\sum_{j=1}^{|V|} P(i,j)log\left(\frac{P(i,j)}{P(i)P'(j)}\right), \tag{C.6}$$

where $P(i,j) = |U_i \cap V_j|/N$ is the probability that an object picked at random falls into both classes U_i and V_j.

The Normalised Mutual Information is defined as:

$$NMI(U,V) = \frac{MI(U,V)}{\sqrt{H(U)H(V)}}. \tag{C.7}$$

The equations here are taken from [Vinh et al., 2009].

Bibliography

J. Aggarwal and L. Xia. Human activity recognition from 3D data: A review. *Pattern Recognition Letters*, 48:70 – 80, 2014. 10, 17

J. K. Aggarwal and M. S. Ryoo. Human activity analysis: A review. *ACM Computing Surveys (CSUR)*, 43(3):16, 2011. 36

P. E. Agre and D. Chapman. Pengi: An implementation of a theory of activity. In *Proc. Association for the Advancement of Artificial Intelligence (AAAI)*, 1987. 35

J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983. vii, 19, 69, 71

M. Alomari, P. Duckworth, N. Bore, M. Hawasly, D. C. Hogg, and A. G. Cohn. Grounding of human environments and activities for autonomous robots. In *26th International Joint Conference on Artificial Intelligence (IJCAI)*, 2017a. 29, 119

M. Alomari, P. Duckworth, D. Hogg, and A. Cohn. Semi-supervised natural language acquisition and grounding for robotic systems. In *Proc. Association for the Advancement of Artificial Intelligence (AAAI)*, 2017b. 24, 28, 156

P. X. Amorapanth, P. Widick, and A. Chatterjee. The neural basis for spatial relations. *Journal of Cognitive Neuroscience*, 22(8):1739–1753, 2010. 18

V. Argyriou, M. Petrou, and S. Barsky. Photometric stereo with an arbitrary number of illuminants. *Computer Vision and Image Understanding*, 114(8):887–900, 2010. 10

K. Arras, O. Mozos, and W. Burgard. Using boosted features for the detection of people in 2d range data. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2007. 45

A. Asuncion, M. Welling, P. Smyth, and Y. W. Teh. On smoothing and inference for topic models. In *Proc. of Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 27–34. AUAI Press, 2009. 147, 150

A. Basharat, A. Gritai, and M. Shah. Learning object motion patterns for anomaly detection and improved object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. 16

A. Behera, A. Cohn, and D. Hogg. Workflow activity monitoring using dynamics of pair-wise qualitative spatial relations. *Advances in Multimedia Modeling*, pages 196–209, 2012a. 24, 124, 156

A. Behera, D. C. Hogg, and A. G. Cohn. Egocentric activity monitoring and recovery. In *Asian Conference on Computer Vision (ACCV)*, 2012b. 22, 28, 154

N. Bellotto and H. Hu. Multisensor-based human detection and tracking for mobile service robots. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(1): 167–181, 2009. 45

M. Bennewitz, W. Burgard, and S. Thrun. Using EM to learn motion behaviors of persons with mobile robots. In *IEEE Conference on Intelligent Robots and Systems (IROS)*, 2002. 26, 114

M. Bennewitz, W. Burgard, G. Cielniak, and S. Thrun. Learning motion patterns of people for compliant robot motion. *International Journal of Robotics Research*, 24:31–48, 2005. 26

J. Bischof and E. M. Airoldi. Summarizing topical content with word frequency and exclusivity. In *Proc. of the 29th International Conference on Machine Learning (ICML)*, pages 201–208, 2012. 32

D. Blackwell. Conditional expectation and unbiased sequential estimation. *The Annals of Mathematical Statistics*, pages 105–110, 1947. 98

D. M. Blei and J. D. Lafferty. Dynamic topic models. In *Proc. of the 23rd International Conference on Machine Learning (ICML)*, pages 113–120. ACM, 2006. 101, 157

D. M. Blei, A. Kucukelbir, and J. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*. to appear. 148

D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *Journal of ML research*, 3:993–1022, 2003. 29, 98

G. Bleser, D. Damen, A. Behera, G. Hendeby, K. Mura, M. Miezal, A. Gee, N. Petersen, G. Maçães, H. Domingues, et al. Cognitive learning, monitoring and assistance of industrial workflows using egocentric sensor networks. *PloS one*, 10(6):e0127769, 2015. 28

A. F. Bobick. Movement, activity and action: the role of knowledge in the perception of motion. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 352 (1358):1257–1265, 1997. 3, 11

A. F. Bobick and J. W. Davis. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 23(3):257–267, 2001. 11, 13

P. Bogaert. *A qualitative calculus for moving point objects constrained by networks*. PhD thesis, Ghent University, 2008. 58

L. Bolelli, S. Ertekin, and C. L. Giles. Topic and trend detection in text collections using latent dirichlet allocation. In *European Conference on Information Retrieval (ECIR)*, pages 776–780. Springer, 2009. 29

N. Bore, R. Ambrus, P. Jensfelt, and J. Folkesson. Efficient retrieval of arbitrary objects from long-term robot observations. *Robotics and Autonomous Systems*, 2017. 29, 43

A. Borzin, E. Rivlin, and M. Rudzsky. Surveillance event interpretation using generalized stochastic petri nets. In *Eighth International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, pages 4–4. IEEE, 2007. 14

L. Bottou, Y. Bengio, et al. Convergence properties of the k-means algorithms. *Advances in neural information processing systems (NIPS)*, pages 585–592, 1995. 84

A. Boularias, F. Duvallet, J. Oh, and A. Stentz. Grounding spatial relations for outdoor robot navigation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1976–1982. IEEE, 2015. 28

F. Brémond. *Scene Understanding: perception, multi-sensor fusion, spatio-temporal reasoning and activity recognition*. PhD thesis, HDR Université de Nice-Sophia Antipolis, 2007. 3, 13

B. Bruno, J. Grosinger, F. Mastrogiovanni, F. Pecora, A. Saffiotti, S. Sathyakeerthy, and A. Sgorbissa. Multi-modal sensing for human activity recognition. In *24th International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 594–600. IEEE, 2015. 17

W. Buntine and A. Jakulin. Discrete component analysis. In *Subspace, Latent Structure and Feature Selection*, pages 1–33. Springer, 2006. 29

D. Cao, O. T. Masoud, D. Boley, and N. Papanikolopoulos. Online motion classification using support vector machines. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2291–2296, 2004. 13

G. Casella and E. I. George. Explaining the gibbs sampler. *The American Statistician*, 46(3): 167–174, 1992. 92

F. Castanedo, D. López-de Ipina, H. K. Aghajan, and R. P. Kleihorst. Building an occupancy model from sensor networks in office environments. *5th International ACM/IEEE Conference on Distributed Smart Cameras (ICDSC)*, 3:1–6, 2011. 29

F. Castanedo, D. L. de Ipiña, H. K. Aghajan, and R. Kleihorst. Learning routines over long-term sensor data using topic models. *Expert Systems*, 31(4):365–377, 2014. 29

R. B. Cattell. The scree test for the number of factors. *Multivariate Behavioral Research*, 1(2): 245–276, 1966. 91

A. J.-B. Chaney and D. M. Blei. Visualizing topic models. In *International AAAI Conference on Web and Social Media (ICWSM)*, 2012. 32

J. Chang, J. L. Boyd-Graber, S. Gerrish, C. Wang, and D. M. Blei. Reading tea leaves: How humans interpret topic models. In *Advances in neural information processing systems (NIPS)*, volume 31, pages 1–9, 2009. 32, 100

J. Charles, T. Pfister, D. Magee, D. Hogg, and A. Zisserman. Upper body pose estimation with temporal sequential forests. In *Proc. of the British Machine Vision Conference (BMVC)*, 2014. 12

S. H. Chavoshi, B. De Baets, Y. Qiang, G. De Tré, T. Neutens, and N. Van de Weghe. A qualitative approach to the identification, visualisation and interpretation of repetitive motion patterns in groups of moving point objects. *International Arab Journal of Information Technology*, 12(5):415–423, 2015. 23

J. Chen, A. Cohn, D. Liu, S. Wang, J. Ouyang, and Q. Yu. A survey of qualitative spatial representations. *The Knowledge Engineering Review*, 30:106–136, 2015. v, 4, 18, 19, 20, 21, 54, 66

L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, and Z. Yu. Sensor-based activity recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6):790–808, 2012. 9

Y. Chen, L. Wang, M. Dong, and J. Hua. Exemplar-based visualization of large document corpus. *IEEE Transactions on Visualization and Computer Graphics*, 15(6), 2009. 32

Y. Chen, T. Diethe, and P. Flach. Adl: A topic model for recognition of activities of daily living in a smarthome. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI)*, 2016. 17

A. Chrungoo, S. Manimaran, and B. Ravindran. Activity recognition for natural human robot interaction. In *Social Robotics*, pages 84–94. Springer, 2014. 22, 27

G. Cielniak, M. Bennewitz, and W. Burgard. Where is ...? learning and utilizing motion patterns of persons with mobile robots. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2003. 26

E. Cippitelli, S. Gasparrini, E. Gambi, and S. Spinsante. A human activity recognition system using skeleton data from rgbd sensors. *Computational intelligence and neuroscience*, page 21, 2016. 17

E. Clementini, P. Di Felice, and D. Hernández. Qualitative representation of positional information. *Artificial Intelligence*, 95(2):317 – 356, 1997. 57, 59

A. G. Cohn. Calculi for qualitative spatial reasoning. In *Artificial Intelligence and Symbolic Mathematical Computation*, pages 124–143. Springer, 1996. 19

A. G. Cohn, B. Bennett, J. Gooday, and N. M. Gotts. Representing and reasoning with qualitative spatial relations about regions. In *Spatial and temporal reasoning*, pages 97–134. Springer, 1997. 19

A. G. Cohn, D. R. Magee, A. Galata, D. C. Hogg, and S. M. Hazarika. Towards an architecture for cognitive vision using qualitative spatio-temporal representations and abduction. In *Spatial cognition III*, pages 232–248. Springer, 2003. 3

A. G. Cohn, J. Renz, and M. Sridhar. Thinking inside the box: A comprehensive spatial representation for video analysis. In *International Conference on Principles of Knowledge Representation and Reasoning (KR)*, 2012. 12

A. G. Cohn, S. Li, W. Liu, and J. Renz. Reasoning about topological and cardinal direction relations between 2-dimensional spatial objects. *Journal of Artificial Intelligence Research (JAIR)*, 51:493–532, 2014. 18

J.-F. Condotta, M. Saade, and G. Ligozat. A generic toolkit for n-ary qualitative temporal and spatial calculi. In *13th International Symposium on Temporal Representation and Reasoning*, pages 78–86. IEEE, 2006. 25

N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 886–893, 2005. 11

B. de Finetti. Theory of probability: A critical introductory treatment. vol. 6. Reprint of the 1975 translation, 1990. 94, 156

H. N. de Ridder et al. Information System on Graph Classes and their Inclusions (ISGCI). www.graphclasses.org (Interval Graphs), 2016. 70

H. Dee and D. Hogg. Detecting inexplicable behaviour. In *Proc. of the British Machine Vision Conference (BMVC)*, 2004. 16

S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6): 391, 1990. 29, 86

M. Delafontaine, A. G. Cohn, and N. Van de Weghe. Implementing a qualitative calculus to analyse moving point objects. *Expert Systems with Applications*, 38(5):5187 – 5196, 2011. 23, 58

M. Devanne, S. Berretti, P. Pala, H. Wannous, M. Daoudi, and A. Del Bimbo. Motion segment decomposition of rgb-d sequences for human behavior understanding. *Pattern Recognition*, 61:222–233, 2017. 18

J. M. Dickey, J. M. Jiang, and J. B. Kadane. Bayesian methods for censored categorical data. *Journal of the American Statistical Association*, 82(399):773–781, 1987. 98

P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 65–72. IEEE, 2005. 11

J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 13

C. Dondrup, N. Bellotto, and M. Hanheide. Social distance augmented qualitative trajectory calculus for human-robot spatial interaction. In *The 23rd IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 519–524. IEEE, 2014. 23, 25, 27, 58, 66

C. Dondrup, N. Bellotto, F. Jovan, and M. Hanheide. Real-time multisensor people tracking for human-robot spatial interaction. In *Workshop on Machine Learning for Social Robotics at ICRA*, 2015. v, 46

K. Dubba, M. Bhatt, F. Dylla, D. C. Hogg, and A. G. Cohn. Interleaved inductive-abductive reasoning for learning complex event models. In *International Conference on Inductive Logic Programming*, pages 113–129. Springer, 2011. 20

K. Dubba, A. Cohn, D. Hogg, M. Bhatt, and F. Dylla. Learning relational event models from video. *Journal of Artificial Intellignce Research (JAIR)*, 53:41–90, 2015. 12, 14

K. S. R. Dubba, A. G. Cohn, and D. C. Hogg. Event model learning from complex videos using ILP. In *European Conference on Artificial Inteligence (ECAI)*, 2010. 14, 20

P. Duckworth, M. Alomari, Y. Gatsoulis, D. C. Hogg, and A. G. Cohn. Unsupervised activity recognition using latent semantic analysis on a mobile robot. In *22nd European Conference on Artificial Inteligence (ECAI)*, 2016a. 67, 104

P. Duckworth, Y. Gatsoulis, F. Jovan, D. C. Hogg, and A. G. Cohn. Unsupervised learning of qualitative motion behaviours by a mobile robot. In *Proc. of the 15th International Conference on Autonomous Agents & Multiagent Systems (AAMAS)*, 2016b. 67, 72, 103

P. Duckworth, M. Alomari, J. Charles, D. C. Hogg, and A. G. Cohn. Latent dirichlet allocation for unsupervised activity analysis on an autonomous mobile robot. In *Proc. of Association for the Advancement of Artificial Intelligence (AAAI)*, 2017. 104

M. J. Egenhofer and R. D. Franzosa. Point-set topological spatial relations. *International Journal of Geographical Information System*, 5(2):161–174, 1991. v, 19, 21

B. A. El-Geresy and A. I. Abdelmoty. Sparqs: a qualitative spatial reasoning engine. *Knowledge-Based Systems*, 17(2):89–102, 2004. 25

J. Fernyhough, A. G. Cohn, and D. C. Hogg. Building qualitative event models automatically from visual input. In *IEEE International Conference on Computer Vision (ICCV)*, pages 350–355, 1998. v, 12, 18, 20, 23, 24, 58, 66

J. Fernyhough, A. G. Cohn, and D. C. Hogg. Constructing qualitative event models automatically from video input. *Image and Vision Computing*, 18(2):81–103, 2000. 66

J. H. Fernyhough. *Generation of qualitative spatio-temporal representations from visual input.* PhD thesis, The University of Leeds, 1997. 66

K. D. Forbus. Elementary school science as a cognitive system domain: How much qualitative reasoning is required? *Advances in Cognitive Systems*, 2016. 19

A. U. Frank. Qualitative spatial reasoning about distances and directions in geographic space. *Journal of Visual Languages & Computing*, 3(4):343–371, 1992. 22

A. U. Frank. Qualitative spatial reasoning: Cardinal directions as an example. *International Journal of Geographical Information Science*, 10(3):269–290, 1996. v, 21, 22

P. Frasconi, F. Costa, L. De Raedt, and K. De Grave. klog: A language for logical and relational learning with kernels. *Artificial Intelligence*, 217:117–143, 2014. 79

A. Galata, A. Cohn, D. Magee, and D. Hogg. Modeling interaction using learnt qualitative spatio-temporal relations and variable length markov models. In *European Conference on Artificial Intelligence (ECAI)*, pages 741–745, 2002. 24

Y. Gatsoulis, M. Alomari, C. Burbridge, C. Dondrup, P. Duckworth, P. Lightbody, M. Hanheide, N. Hawes, and A. G. Cohn. QSRlib: a software library for online acquisition of Qualitative Spatial Relations from Video. In *Workshop on Qualitative Reasoning, at IJCAI*, 2016a. 5, 25, 58, 67, 79

Y. Gatsoulis, P. Duckworth, C. Dondrup, P. Lightbody, and C. Burbridge. QSRlib: A library for qualitative spatial-temporal relations and reasoning. qsrlib.readthedocs.org, Jan 2016b. 5, 58, 67, 79

A. Gelman, J. Carlin, H. Stern, and D. Rubin. *Bayesian data analysis*, volume 2. Chapman & Hall/CRC Boca Raton, FL, USA, 2014. 148

S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence (PAMI)*, (6):721–741, 1984. 92

L. Gorelick, M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1395–1402, 2005. 12, 13

I. Gori, J. Sinapov, P. Khante, P. Stone, and J. K. Aggarwal. Robot-centric activity recognition "in the wild". In *Social Robotics*, pages 224–234. Springer International Publishing, 2015. 28

D. Govindaraju and M. Veloso. Learning and recognizing activities in streams of video. In *Workshop on Learning in Computer Vision, at AAAI*, 2005. 27

R. K. Goyal and M. J. Egenhofer. Similarity of cardinal directions. In *Advances in Spatial and Temporal Databases*, pages 36–55. Springer, 2001. 22

T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National academy of Sciences*, 101(suppl 1):5228–5235, 2004. 98, 99

G. Grisetti, C. Stachniss, and W. Burgard. Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE transactions on Robotics*, 23(1):34–46, 2007. 39

F. Gu, M. Sridhar, A. Cohn, D. Hogg, F. Flórez-Revuelta, D. Monekosso, and P. Remagnino. Weakly supervised activity analysis with spatio-temporal localisation. *Neurocomputing*, 216: 778–789, 2016. 15

A. Gupta, A. Shafaei, J. J. Little, and R. J. Woodham. Unlabelled 3d motion examples improve cross-view action recognition. 2014. 17

N. Halko, P. G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011. 89

R. Hamid, S. Maddi, A. Johnson, A. Bobick, I. Essa, and C. Isbell. A novel sequence representation for unsupervised analysis of human activities. *Artificial Intelligence*, 173(14):1221–1244, 2009. 36

J. Han, L. Shao, D. Xu, and J. Shotton. Enhanced computer vision with microsoft kinect sensor: A review. *IEEE Transactions on Cybernetics*, 43(5):1318–1334, 2013. 12

A. Haque, B. Peng, Z. Luo, A. Alahi, S. Yeung, and L. Fei-Fei. Towards viewpoint invariant 3d human pose estimation. In *European Conference on Computer Vision*, pages 160–177. Springer, 2016. 13

C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Manchester, UK, 1988. 11

N. Hawes, M. Klenk, K. Lockwood, G. S. Horn, and J. D. Kelleher. Towards a cognitive system that can recognize spatial regions based on context. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2012. 28

N. Hawes, C. Burbridge, F. Jovan, L. Kunze, B. Lacerda, L. Mudrová, J. Young, J. L. Wyatt, D. Hebesberger, T. Körtner, R. A. N. Bore, J. Folkesson, P. Jensfelt, L. Beyer, A. Hermans, B. Leibe, A. Aldoma, T. Faulhammer, M. Z. M. Vincze, M. Al-Omari, E. Chinellato, P. Duckworth, Y. Gatsoulis, D. C. Hogg, A. G. Cohn, C. Dondrup, J. P. Fentanes, T. Krajník, J. M. Santos, T. Duckett, and M. Hanheide. The STRANDS project: Long-term autonomy in everyday environments. *IEEE Robotics and Automation Magazine*, In Press, 2016. 2, 26, 38

J. Hertzberg, J. Zhang, L. Zhang, S. Rockel, B. Neumann, J. Lehmann, K. Dubba, A. Cohn, A. Saffiotti, F. Pecora, and others. The race project. *KI-Künstliche Intelligenz*, 28(4):297–304, 2014. 26

M. Hoffman, F. Bach, and D. Blei. Online learning for latent dirichlet allocation. In *Advances in neural information processing systems (NIPS)*, 2010. 147, 148, 168

T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine learning*, 42(1-2):177–196, 2001. 29

L. Hong and B. D. Davison. Empirical study of topic modeling in twitter. In *Proceedings of the first workshop on social media analytics*, pages 80–88. ACM, 2010. 32

S. Hongeng. Unsupervised learning of multi-object events. In *Proc. of the British Machine Vision Conference BMVC*, pages 1–10, 2004. 14

S. Hongeng and R. Nevatia. Multi-agent event recognition. In *IEEE International Conference on Computer Vision (ICCV)*, pages 84–91, 2001. 12, 14

S. Hongeng and R. Nevatia. Large-scale event detection using semi-hidden markov models. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1455–1462, 2003. 14

N. Hu, G. Englebienne, Z. Lou, and B. Kröse. Learning latent structure for activity recognition. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1048–1053. IEEE, 2014. 18

W. Hu, D. Xie, and T. Tan. A hierarchical self-organizing approach for learning the patterns of motion trajectories. *IEEE Trans. on Neural Networks*, 15:135–144, 2004. 16

W. Hu, X. Xiao, Z. Fu, D. Xie, and T. Tan. A system for learning statistical motion patterns. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 28:1450–1464, 2006. 16

M. E. Hussein, M. Torki, M. A. Gowayyed, and M. El-Saban. Human action recognition using a temporal hierarchy of covariance descriptors on 3D joint locations. In *International Joint Conference on Artificial Inteligence (IJCAI)*, 2013. 27

Y. A. Ivanov and A. F. Bobick. Recognition of visual activities and interactions by stochastic parsing. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 22(8): 852–872, 2000. 10

K.-H. Jo, Y. Kuno, and Y. Shirai. Manipulative hand gesture recognition using task knowledge for human computer interaction. In *Proc. of the 3rd IEEE International Conference on Automatic Face and Gesture Recognition*, pages 468–473. IEEE, 1998. 13

N. Johnson and D. Hogg. Learning the distribution of object trajectories for event recognition. In *Proc. of the British Machine Vision Conference (BMVC)*, 1995. 16

E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python, 2001. URL `http://www.scipy.org/`. [Online; accessed ¡today¿]. 121

M. I. Jordan. *Learning in graphical models*, volume 89. Springer Science & Business Media, 1998. 98

T. Kanda, D. Glas, M. Shiomi, and N. Hagita. Abstracting people's trajectories for social robots to proactively approach customers. *IEEE Trans on Robotics*, 25:1382–1396, 2009. 26

Kinect camera [online]. Available: http://www.xbox.com/en-US/kinect/default.htm, 2017. 10, 12

D. Kirsh. The intelligent use of space. *Artificial intelligence*, 73(1-2):31–68, 1995. 36

K. Kitani, B. Ziebart, J. Bagnell, and M. Hebert. Activity forecasting. *European Conference on Computer Vision ECCV*, pages 201–214, 2012. 16

H. Koppula and A. Saxena. Learning spatio-temporal structure from rgb-d videos for human activity detection and anticipation. In *Proc. of the International Conference on Machine Learning (ICML)*, pages 792–800, 2013. 18

L. Kunze and N. Hawes. Soma project report. STRANDS project report. In preperation., 2017. 44

L. Kunze, C. Burbridge, M. Alberti, A. Thippur, J. Folkesson, P. Jensfelt, and N. Hawes. Combining top-down spatial reasoning and bottom-up object class recognition for scene understanding. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2014. 23, 25

J. Lafferty, A. McCallum, F. Pereira, et al. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001. 14

I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2-3): 107–123, 2005. 11

I. Laptev and P. Pérez. Retrieving actions in movies. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1–8, 2007. 13

I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008. 11

O. D. Lara and M. A. Labrador. A survey on human activity recognition using wearable sensors. *IEEE Communications Surveys and Tutorials*, 15(3):1192–1209, 2013. 9

F. Large, D. Vasquez, T. Fraichard, and C. Laugier. Avoiding cars and pedestrians using velocity obstacles and motion prediction. In *Intelligent Vehicles Symposium*, pages 375–379. IEEE, 2004. 27

G. Lavee, E. Rivlin, and M. Rudzsky. Understanding video events: a survey of methods for automatic interpretation of semantic occurrences in video. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews,*, 39(5):489–504, 2009. 3, 10, 13, 17, 36

L. Liao, D. J. Patterson, D. Fox, and H. Kautz. Learning and inferring transportation routines. *Artificial Intelligence*, 171:311–331, 2007. 16

J. Liu, S. Ali, and M. Shah. Recognizing human actions using multiple features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. 30

D. F. Llorca, V. Milanes, I. P. Alonso, M. Gavilan, I. G. Daza, J. Perez, and M. . Sotelo. Autonomous pedestrian collision avoidance using a fuzzy steering controller. *IEEE Transactions on Intelligent Transportation Systems*, 12(2):390–401, 2011. 27

M. Luber, L. Spinello, J. Silva, and K. Arras. Socially-aware robot navigation: A learning approach. In *IEEE Conference on Intelligent Robots and Systems (IROS)*, 2012. 16

F. Lv and R. Nevatia. Single view human action recognition using key pose matching and viterbi path searching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2007. 13

S. M. Lynch. *Introduction to applied Bayesian statistics and estimation for social scientists.* Springer Science & Business Media, 2007. 98

J. MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967. 81

C. D. Manning, . Raghavan, H. Schütze, et al. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008. 88

E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey, and K. Konolige. The office marathon. In *IEEE Conference on Robotics and Automation (ICRA)*, 2010. 2, 25

G. Medioni, I. Cohen, F. Brémond, S. Hongeng, and R. Nevatia. Event detection and analysis from video streams. *IEEE Transactions on pattern analysis and machine intelligence*, 23(8): 873–889, 2001. 10

MetraLabs. www.metralabs.com/en, 2017. 26, 38

Microsoft Xbox. http://www.xbox.com, 2017. 12

D. Mitzel and B. Leibe. Close-range human detection for head-mounted cameras. In *Proc. of the British Machine Vision Conference BMVC*, pages 1–11, 2012. 45

M. Mohnhaupt and B. Neumann. Understanding object motion: Recognition, learning and spatiotemporal reasoning. *Robotics and Autonomous Systems*, 8(1):65 – 91, 1991. Special Issue Toward Learning Robots. 24

V. I. Morariu and L. S. Davis. Multi-agent event recognition in structured scenarios. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3289–3296. IEEE, 2011. 10

R. Moratz and M. Ragni. Qualitative spatial reasoning about relative point position. *Journal of Visual Languages & Computing*, 19(1):75–98, 2008. 57, 61

S. Moyle and S. Muggleton. Learning programs in the event calculus. *Inductive Logic Programming*, pages 205–212, 1997. 14

J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1):32–38, 1957. 143

J. C. Niebles and L. Fei-Fei. A hierarchical model of shape and appearance for human action classification. In *Conference on Computer Vision and Pattern Recognition, CVPR*, pages 1–8. IEEE, 2007. 154

J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *International Journal of Computer Vision*, 79(3):299–318, 2008. 14, 29, 30

T. Oates. PERUSE: An unsupervised algorithm for finding recurring patterns in time series. In *International Conference on Data Mining, (ICDM)*, pages 330–337. IEEE, 2002. 26

A. Ogale, A. Karapurkar, G. Guerra-Filho, and Y. Aloimonos. View-invariant identification of pose sequences for action recognition. In *Video Analysis and Content Extraction Workshop (VACE)*, 2004. 14

OpenNI organization. www.openni.org/, 2016. 12, 48

J. Papon, A. Abramov, M. Schoeler, and F. Worgotter. Voxel cloud connectivity segmentation-supervoxels for point clouds. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 43

G. I. Parisi, C. Weber, and S. Wermter. Self-organizing neural integration of pose-motion features for human action recognition. *Frontiers in neurorobotics*, 9, 2015. 17

H. Pfister, M. Zwicker, J. Van Baar, and M. Gross. Surfels: Surface elements as rendering primitives. In *Computer Graphics and Interactive Techniques*, 2000. 41

T. Pfister, J. Charles, and A. Zisserman. Flowing convnets for human pose estimation in videos. In *International Conference on Computer Vision*, 2015. 12

C. Piciarelli, G. L. Foresti, and L. Snidaro. Trajectory clustering and its applications for video surveillance. In *IEEE Conference on Advanced Video and Signal Based Surveillance*, 2005. 16

PR2 Robot Platform. http://wiki.ros.org/Robots/PR2, 2017. 25

P. Quelhas, F. Monay, J. M. Odobez, D. Gatica-Perez, and T. Tuytelaars. A thousand words in a scene. *IEEE transactions on pattern analysis and machine intelligence (PAMI)*, 29(9), 2007. 31

M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng. ROS: an open-source Robot Operating System. *Workshop on open source software at ICRA*, 3(3.2):5, 2009. 12, 39

D. D. M. Ranasinghe and A. S. Karunananda. Qualitative reasoning engine for visual scene understanding in cognitive vision systems. In *International Conference on Information and Automation*, 2006. 22

D. A. Randell, Z. Cui, and A. G. Cohn. A spatial logic based on regions and connection. *Conference on Knowledge Representation and Reasoning (KR)*, 92:165–176, 1992. xi, 19

D. Ravi, C. Wong, B. Lo, and G.-Z. Yang. Deep learning for human activity recognition: A resource efficient implementation on low-power devices. In *Wearable and Implantable Body Sensor Networks (BSN), 2016 IEEE 13th International Conference on*, pages 71–76. IEEE, 2016. 10

D. J. Rezende, S. A. Eslami, S. Mohamed, P. Battaglia, M. Jaderberg, and N. Heess. Unsupervised learning of 3d structure from images. In *Advances in Neural Information Processing Systems (NIPS*, pages 4996–5004, 2016. 31

ROS gmapping. http://wiki.ros.org/gmapping, 2017. 39

ROS msgs. http://wiki.ros.org/msgs, 2017. 161

A. Rosenberg and J. Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *EMNLP-CoNLL*, 2007. 125

P. W. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987. 82

M. Ryoo, T. J. Fuchs, L. Xia, J. K. Aggarwal, and L. Matthies. Robot-centric activity prediction from first-person videos: What will they do to me? In *Proc. of the 10th Annual International Conference on Human-Robot Interaction*, pages 295–302. ACM, 2015. 28

M. S. Ryoo and J. K. Aggarwal. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, 2009. 10

P. P. San, P. Kakar, X.-L. Li, S. Krishnaswamy, J.-B. Yang, and M. N. Nguyen. *Deep Learning for Human Activity Recognition (Chapter 9)*, pages 186 – 204. Intelligent Data-Centric Systems. Academic Press, 2017. 10

S. Savarese, A. DelPozo, J. C. Niebles, and L. Fei-Fei. Spatial-temporal correlatons for unsupervised action classification. In *IEEE Workshop on Motion and video Computing (WMVC)*, 2008. 11, 30

J. Schlenzig, E. Hunter, and R. Jain. Recursive identification of gesture inputs using hidden markov models. In *Proc. of the Second IEEE Workshop on Applications of Computer Vision*, pages 187–194. IEEE, 1994. 14

M. Schoeler, J. Papon, and F. Worgotter. Constrained planar cuts-object partitioning for point clouds. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 43

C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *IEEE International Conference on Pattern Recognition (ICPR)*, pages 32–36, 2004. 11

G. Schwarz et al. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978. 82

Y. Sheikh, M. Sheikh, and M. Shah. Exploring the space of a human action. In *International Conference on Computer Vision. (ICCV)*, volume 1, pages 144–149. IEEE, 2005. 11, 18

Y. Shi, A. Bobick, and I. Essa. Learning temporal sequence model from partially labeled data. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 1631–1638. IEEE, 2006. 16

C. Sievert and K. E. Shirley. Ldavis: A method for visualizing and interpreting topics. In *Proceedings of the workshop on interactive language learning, visualization, and interfaces*, pages 63–70, 2014. 32

J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering objects and their location in images. In *10th IEEE International Conference on Computer Vision (ICCV)*, volume 1, pages 370–377. IEEE, 2005. 30

T. Southey and J. J. Little. Object discovery through motion, appearance and shape. 2006. 29

T. Southey and J. J. Little. Learning qualitative spatial relations for object classification. 2007. 23, 24

T. Southey and J. J. Little. 3d spatial relationships for improving object detection. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2013. 23

Spatio-Temporal Representations and Activities for Cognitive Control in Long-Term Scenarios STRANDS project. strands-project.eu, 2017. 39

M. Sridhar. *Unsupervised learning of event and object classes from video.* PhD thesis, The University of Leeds, 2010. 13, 31, 72

M. Sridhar, A. G. Cohn, and D. C. Hogg. Learning functional object categories from a relational spatio-temporal representation. In *European Conference on Artificial Intelligence (ECAI)*, pages 606–610. IOS Press, 2008. 37, 156

M. Sridhar, A. G. Cohn, and D. C. Hogg. Unsupervised learning of event classes from video. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2010. 12, 20, 31, 54, 154

M. Stikic, K. Van Laerhoven, and B. Schiele. Exploring semi-supervised and active learning for activity recognition. In *12th IEEE international symposium on Wearable computers (ISWC)*, pages 81–88. IEEE, 2008. 16

J. Sung, C. Ponce, B. Selman, and A. Saxena. Unstructured human activity detection from rgbd images. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 842–849. IEEE, 2012. 17, 27

J. Sung, H. Koppula, B. Selman, and A. Saxena. Cornell activity datasets: CAD-60 & CAD-120, http://pr.cs.cornell.edu/humanactivities/, Jun 2014. URL `http://pr.cs.cornell.edu/humanactivities/`. viii, 116, 117

J. Tayyub, A. Tavanai, Y. Gatsoulis, A. Cohn, and D. Hogg. Qualitative and quantitative spatio-temporal relations in daily living activity recognition. In *12th Asian Conference on Computer Vision ACCV*, 2015. 18, 20

A. Thippur, C. Burbridge, L. Kunze, M. Alberti, J. Folkesson, P. Jensfelt, and N. Hawes. A comparison of qualitative and metric spatial relation models for scene understanding. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2015. 25

S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics*. MIT press, 2005. 25

D. Trabelsi, S. Mohammed, F. Chamroukhi, L. Oukhellou, and Y. Amirat. An unsupervised approach for automatic activity recognition based on hidden markov model regression. *IEEE Transactions on Automation Science and Engineering*, 10(3):829–835, 2013. 10

P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea. Machine recognition of human activities: A survey. *IEEE Trans. on Circuits and Systems for Video Technology*, 18(11): 1473–1488, 2008. 17

N. Van de Weghe. Representing and reasoning about moving objects: A qualitative approach (volume i). *Department of Geography-Faculty of Sciences. Ghent University, Ghent*, 168, 2004. 57, 58

N. Van de Weghe, A. G. Cohn, and P. D. Maeyer. A qualitative representation of trajectory pairs. In *Proc. of the 16th European Conference on Artificial Intelligence (ECAI)*, pages 1101–1102. IOS Press, 2004. 23, 58

N. Van de Weghe, A. Cohn, P. De Maeyer, and F. Witlox. Representing moving objects in computer-based expert systems: The overtake event example. *Expert Systems with Applications*, 29:977–983, 2005a. 57, 58

N. Van de Weghe, G. De Tré, B. Kuijpers, and P. De Maeyer. The double-cross and the generalization concept as a basis for representing and comparing shapes of polylines. In *On the Move to Meaningful Internet Systems 2005: OTM 2005 Workshops*, pages 1087–1096. Springer, 2005b. 58

N. Van de Weghe, A. Cohn, G. De Tre, and P. De Maeyer. A qualitative trajectory calculus as a basis for representing moving objects in geographical information systems. *Control and Cybernetics*, 35(1):97, 2006. 57, 58

D. Vasquez and T. Fraichard. Motion prediction for moving objects: a statistical approach. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, volume 4, pages 3931–3936. IEEE, 2004. 16

N. X. Vinh, J. Epps, and J. Bailey. Information theoretic measures for clusterings comparison: Is a correction for chance necessary? In *Proc. of the 26th Annual International Conference on Machine Learning (ICML)*, 2009. 125, 169

P. Viswanathan, J. J. Little, A. K. Mackworth, T.-V. How, R. H. Wang, and A. Mihailidis. Intelligent wheelchairs for cognitively-impaired older adults in long-term care: A review. *Rehabilitation engineering and assistive technology society of North America, Bellevue, WA*, 2013. 157

T. Wagner, U. Visser, and O. Herzog. Egocentric qualitative spatial knowledge representation for physical robots. *Robotics and Autonomous Systems*, 49(1):25–42, 2004. 28, 54

H. M. Wallach. Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, pages 977–984. ACM, 2006. 156

J. O. Wallgrün, L. Frommberger, D. Wolter, F. Dylla, and C. Freksa. Qualitative spatial representation and reasoning in the sparq-toolbox. Springer, 2006. 25

J. Wang, Z. Chen, and Y. Wu. Action recognition with multiscale spatio-temporal contexts. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3185–3192, 2011. 11

X. Wang, X. Ma, and W. E. L. Grimson. Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 31(3):539–555, 2009. 31

Y. Wang, E. Agichtein, and M. Benzi. Tm-lda: efficient online modeling of latent topic transitions in social media. In *Proc. of the 18th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 123–131. ACM, 2012. 29

S. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 12, 49

D. Weinland, R. Ronfard, and E. Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding*, 115(2): 224–241, 2011. 17

D. S. Weld and J. De Kleer. *Readings in qualitative reasoning about physical systems*. Morgan Kaufmann, 2013. 57

S. Wong, T. K. Kim, and R. Cipolla. Learning motion categories using both semantic and structural information. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007. 30

L. Xia, I. Gori, J. K. Aggarwal, and M. S. Ryoo. Robot-centric activity recognition from first-person RGB-D videos. In *IEEE Conf. on Applications of Computer Vision (WACV)*, 2015. 27, 28

L. Xie, H. Sundaram, and M. Campbell. Event mining in multimedia streams. *Proceedings of the IEEE*, 96(4):623–647, 2008. 4

Xtion ASUS camera [online]. www.asus.com/3D-Sensor/. 10, 12

J. Yang, J. Wang, and Y. Chen. Using acceleration measurements for activity recognition: An effective learning algorithm for constructing neural classifiers. *Pattern recognition letters*, 29 (16):2213–2220, 2008. 13

X. Yang, C. Zhang, and Y. Tian. Recognizing actions using depth motion maps-based histograms of oriented gradients. In *Proc. of the 20th ACM International Conference on Multimedia*, pages 1057–1060. ACM, 2012. 11

M. Ye, Q. Zhang, L. Wang, J. Zhu, R. Yang, and J. Gall. A survey on human motion analysis from depth data. In *Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications*, pages 149–187. Springer, 2013. 17

J. Young and N. Hawes. Learning by observation using qualitative spatial relations. In *IEEE Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2015. 23, 54, 67

J. Young, V. Basile, L. Kunze, E. Cabrio, and N. Hawes. Towards lifelong object learning by integrating situated robot perception and semantic web mining. In *Proc. of the European Conference on Artificial Intelligence (ECAI)*, 2016. 43

J. Young, L. Kunze, V. Basile, E. Cabrio, N. Hawes, and B. Caputo. Semantic web-mining and deep vision for lifelong object discovery. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2017. 43, 135, 156

L. Zelnik-Manor and M. Irani. Statistical analysis of dynamic actions. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 28(9):1530–1535, 2006. 3, 11

C. Zhang and Y. Tian. RGB-D camera-based daily living activity recognition. *Journal of Computer Vision and Image Processing*, 2(4):12, 2012. 22

D. Zhang, D. Gatica-Perez, S. Bengio, and I. McCowan. Semi-supervised adapted hmms for unusual event detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 611–618. IEEE, 2005. 16

J. Zhang and S. Gong. Action categorization by structural probabilistic latent semantic analysis. *Computer Vision and Image Understanding*, 114(8):857–864, 2010. 30

B. Zheng, Y. Zhao, J. C. Yu, K. Ikeuchi, and S.-C. Zhu. Beyond point clouds: Scene understanding by reasoning geometry and physics. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3127–3134, 2013. 28

Z. Zhou, X. Chen, Y. C. Chung, Z. He, T. X. Han, and J. M. Keller. Activity analysis, summarization, and visualization for indoor human activity monitoring. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1489–1498, 2008. 17

Q. Zhu, M.-C. Yeh, K.-T. Cheng, and S. Avidan. Fast human detection using a cascade of histograms of oriented gradients. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1491–1498. IEEE, 2006. 11

K. Zimmermann and C. Freksa. Qualitative spatial reasoning using orientation, distance, and path knowledge. *Applied intelligence*, 6(1):49–58, 1996. 58